

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 08-227377

(43)Date of publication of application : 03.09.1996

(51)Int.Cl.

G06F 12/06

G06F 12/06

(21)Application number : 07-293749

(71)Applicant : COMPAQ COMPUTER CORP

(22)Date of filing : 16.10.1995

(72)Inventor : STEVENS JEFFREY C

LARSON JOHN E

THOME GARY W

COLLINS MICHAEL J

MORIARTY MICHAEL

(30)Priority

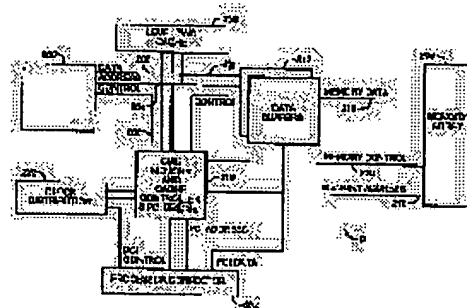
Priority number : 94 323517 Priority date : 14.10.1994 Priority country : US

(54) MEMORY CONTROLLER FOR ACCESS TO MEMORY DEVICES AT DIFFERENT SPEEDS IN DIFFERENT CYCLES

(57)Abstract:

PROBLEM TO BE SOLVED: To operate memory devices at different speeds in different cycles.

SOLUTION: This device is provided with a means for receiving address and cycle timing information, and deciding which bank of a DRAM should be accessed, a means for indicating a DRAM format for each bank, a means for storing plural DRAM timing parameters for the part of 6 DRAM cycle for each DRAM format, and a means for deciding the plural DRAM timing parameters suitable to a received address based on the DRAM format indication and the bank decision for each decided bank. The decided plural DRAM timing parameters are received, and the decided bank and address supplies a column, column address, and address strobe to the DRAM according to the decided plural DRAM timing parameters for the specific bank.



LEGAL STATUS

[Date of request for examination] 17.09.2002

[Date of sending the examiner's decision of rejection] 11.04.2005

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision
of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平8-227377

(43)公開日 平成8年(1996)9月3日

(51)Int.Cl. ⁹	識別記号	庁内整理番号	FI	技術表示箇所
G 0 6 F 12/06	5 1 5		G 0 6 F 12/06	5 1 5 H
	5 2 2			5 2 2 A

審査請求 未請求 請求項の数14 FD 外国語出願 (全 91 頁)

(21)出願番号 特願平7-293749

(22)出願日 平成7年(1995)10月16日

(31)優先権主張番号 08/323, 517

(32)優先日 1994年10月14日

(33)優先権主張国 米国 (US)

(71)出願人 591277326

コンパック・コンピュータ・コーポレイシ
ョン

COMPAQ COMPUTER COR
PORATION

アメリカ合衆国テキサス州77070ヒュース
トン20555エス・エイチ249

(72)発明者 ジェフリー・シー・ステイーヴンズ

アメリカ合衆国77379テキサス州スプリン
グ・ウッドロード・レイン17611

(74)代理人 弁理士 土屋 勝

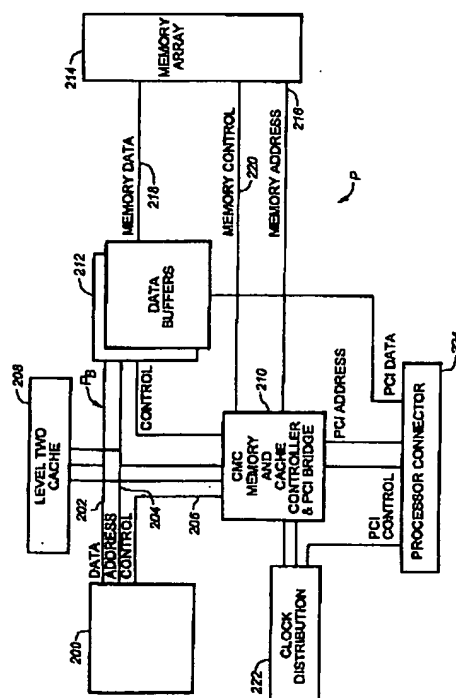
最終頁に続く

(54)【発明の名称】 異なったサイクルで異なる速度のメモリデバイスにアクセスできるメモリ制御器

(57)【要約】 (修正有)

【課題】 異なったサイクルで異なる速度のメモリデバイスを操作する。

【解決手段】 アドレス及びサイクルタイミング情報を受信し、DRAMのどのバンクがアクセスされるべきであるかを決定する手段、各バンク毎にDRAM形式を指示する手段、各DRAM形式毎にDRAMサイクルの部分のための複数のDRAMタイミングパラメータを記憶する手段、並びに決定されたバンク毎にDRAM形式指示及びバンク決定に基づいて受信されたアドレスに適用できる複数のDRAMタイミングパラメータを決定する手段を備え、決定した複数のDRAMタイミングパラメータを受信し、決定したバンク及びアドレスがその特定のバンク用の決定された複数のDRAMタイミングパラメータに従ってDRAMに列及びカラムアドレス及びアドレスストロブを供給する。



1

【特許請求の範囲】

【請求項1】バンクに配置された複数のDRAMと使用されるDRAM制御器であって、このDRAM制御器は、バスからアドレス及びサイクルタイミング情報を受信し、

DRAMのどのバンクが受信されたアドレスのためにアクセスされるべきであるかを決定する手段と、

各バンク毎にDRAM形式を指示する手段と、

各DRAM形式毎にDRAMサイクルの部分のための複数のDRAMタイミングパラメータを記憶する手段と、
決定されたバンク毎にDRAM形式指示及びバンク決定に基づいて前記受信されたアドレスに適用できる複数のDRAMタイミングパラメータを決定する手段と、

決定した複数のDRAMタイミングパラメータを受信し、決定したバンク及びアドレスがその特定のバンク用の決定された複数のDRAMタイミングパラメータに従ってDRAMに列及びカラムアドレス及びアドレスストロブを供給する手段とを備えたDRAM制御器。

【請求項2】DRAM制御器が更にクロック信号を受信し、前記列及びカラムアドレス及びアドレスストロブを供給する手段は、受信されたクロック信号に基づいて同期操作される請求項1に記載のDRAM制御器。

【請求項3】列及びカラムアドレス及びアドレスストロブを供給する前記手段は、ネゲートされたレベルで決定されたバンク以外のDRAMのバンクに列アドレスストロブを供給する請求項1に記載のDRAM制御器。

【請求項4】DRAMがページモードデバイスであって、列及びカラムアドレス及びアドレスストロブを供給する前記手段がページヒット検知器を含み、もしページヒットが検知されるならば、ページモードの列及びカラムアドレス及びアドレスストロブを供給する請求項1に記載のDRAM制御器。

【請求項5】DRAMの各バンクに割り当てられたアドレス範囲を指示する手段を更に備えた請求項1に記載のDRAM制御器。

【請求項6】前記複数のDRAMタイミングパラメータは、ローアドレスストロブからのローアドレスホールド時間、カラムアドレスストロブへのカラムアドレス確立時間、ローアドレスストロブへのローアドレス確立時間及びカラムアドレスストロブ幅を含む請求項1に記載のDRAM制御器。

【請求項7】前記カラムアドレス確立時間パラメータは、読出及び書込操作毎に個々のパラメータを含む請求項6に記載のDRAM制御器。

【請求項8】複数のバンクに配置された複数のDRAMと、

メモリアドレス及びサイクルタイミング情報を供給するプロセッサと、

前記プロセッサのメモリアドレス及びサイクルタイミング情報を転送するバスと、

2

前記バスからアドレス及びサイクルタイミング情報を受信するDRAM制御器とを備え、前記DRAM制御器は、

DRAMのどのバンクが受信されたアドレスのためにアクセスされるべきであるかを決定する手段と、

各バンク毎にDRAM形式を指示する手段と、

各DRAM形式毎にDRAMサイクルの部分のための複数のDRAMタイミングパラメータを記憶する手段と、
決定されたバンク毎にDRAM形式指示及びバンク決定

に基づいて前記受信されたアドレスに適用できる複数のDRAMタイミングパラメータを決定する手段と、

決定した複数のDRAMタイミングパラメータを受信し、決定したバンク及びアドレスがその特定のバンク用の決定された複数のDRAMタイミングパラメータに従ってDRAMに列及びカラムアドレス及びアドレスストロブを供給する手段とを含むコンピュータシステム。

【請求項9】クロック信号を供給する手段を更に備え、前記DRAM制御器が前記クロック信号を受信し、前記列及びカラムアドレス及びアドレスストロブを供給する前記DRAM制御器手段は、受信されたクロック信号に基づいて同期操作される請求項8に記載のコンピュータシステム。

【請求項10】列及びカラムアドレス及びアドレスストロブを供給する前記DRAM制御器手段は、ネゲートされたレベルで決定されたバンク以外のDRAMのバンクに列アドレスストロブを供給する請求項8のコンピュータシステム。

【請求項11】DRAMがページモードデバイスであって、列及びカラムアドレス及びアドレスストロブを供給する前記DRAM制御器手段がページヒット検知器を含み、もしページヒットが検知されるならば、ページモードの列及びカラムアドレス及びアドレスストロブを供給する請求項8のコンピュータシステム。

【請求項12】前記DRAM制御器がDRAMの各バンクに割り当てられたアドレス範囲を指示する手段を更に備えた請求項8のコンピュータシステム。

【請求項13】前記複数のDRAMタイミングパラメータは、ローアドレスストロブからのローアドレスホールド時間、カラムアドレスストロブへのカラムアドレス確立時間、ローアドレスストロブへのローアドレス確立時間及びカラムアドレスストロブ幅を含む請求項8のコンピュータシステム。

【請求項14】前記カラムアドレス確立時間パラメータは、読出及び書込操作毎に個々のパラメータを含む請求項13に記載のコンピュータシステム。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、コンピュータシステムに用いられたメモリ制御器に関し、特に別々のバンク及び追加的に異なる速度のマイクロプロセッサで異なる

る速度のメモリデバイスを操作することができるメモリ制御器に関する。

【0002】

【従来の技術】 コンピュータシステムは日増しにより強力になっている。ユーザはかつてより複雑な知的な応用例を動作させる多くの能力を要求し、コンピュータシステム製造業者がこれに答えている。コンピュータ速度はここ数年に亘って劇的に増加して、今のデスクトップ及びファイル・サーバコンピュータが10～15年前のメインフレームコンピュータを容易に卓越することができる。しかし、更なる性能の探求が決して終わっていない。この結果、マイクロプロセッサ製造業者がより速いマイクロプロセッサを開発している。

【0003】 しかしながら、コンピュータシステムは、マイクロプロセッサのみではない。完全なコンピュータシステムを形成するためには、マイクロプロセッサと協働しなければならない多くの他のサブシステムがある。さまざまなユーザの要望を満足させるためには、これらのサブシステムをできるのみ多く最適化し、それなのにコスト及びシステムの柔軟性を考慮に入れることが望ましい。

【0004】 マイクロプロセッサの開発との歩調が維持できなかったサブシステムの2つは主メモリシステム及びI/Oバスである。主メモリシステムの短所は、キャッシュメモリシステムの使用によって大いに軽減された。しかし結局全てのメモリ操作は、その性能がコンピュータシステムの全体の性能でまだ重要な部分であるように、最終的に主メモリから来なければならない。多くの先進のメモリアーキテクチャ及び技術が長年の間に開発された。最も普通の技術の1つは、実際のメモリアドレス位置値が列(ロー)とカラムに分けられたページモードメモリデバイス即ちDRAMの使用であり、もし列アドレス即ちページが次の操作で同じであるならば、カラムアドレスのみがDRAMに供給される必要がある。必要とされる少しのオーバーヘッドが、ページヒット間に得られた改善された性能によって容易に補える。それで基本的ページモード操作は、主な性能増加を形成するが、高性能はいつも望まれる。

【0005】 1つの更なる性能増加は、メモリシステムが動いていない時に、列アドレスストロブ即ちRAS*信号のレベルを決定するための改善に関する。公知なように、RAS*信号はネゲートされ即ちハイにセットされて新しいページ即ちローアドレスが供給できなければならない、またプリチャージ時間も必要である。従って、次の操作が実際にページヒットである時にもしRAS*信号がハイに上げられたならば、性能ペナルティがある。同様にフルなプリチャージ時間がサイクルが出された後に実行しなければならない時において、もしRAS*信号がローであり、操作がページミスであるならば、遅延がある。この関心を指摘するために種々の技術

は、新しいページサイクルを指示するために、RAS*信号がローに保持されるべきか或いはハイに戻すべきかを予測するために開発された。予測は、1993年3月22日に出願された連番08/034, 104号の表題『ページミスを動的に予測するメモリ制御器』に示されるように、種々の方法で実行することができる。この出願でいくつかの技術が使用された。第1に、単純な技術がいつも固定的な選択によるプロセッサによって行なわれた最終サイクルの形式予測に基づいている。第2に、より知的な技術は、ヒット及びミスを各サイクル型毎にサンプルして、その後この適合的測定に基づいてRAS*レベルをセットする。しかし、この技術はプロセッササイクルの使用に基づいたが、I/Oバスサイクル自身に基づかなかった。それ故、I/Oバスマスタ操作はより低いレベルでまだ機能している。

【0006】 1つの高性能I/Oバスは、インテル社によって開発され、多くのコンピュータ製造業者に受け入れられたPCI即ち周辺部品相互接続バスである。PCIは高性能のバスであって、多数のバスマスタが同席するのを許容する。バスマスタは、必須的に特定業務を行なうが汎用処理業務を行わないローカル・プロセッサである。主プロセッサは、これらのバスマスタを持つことによって、種々の特殊処理タスクの負荷を軽減することができ、より多くのタスクが平行して実施することができ、それによってコンピュータシステムの性能を増加させている。これは、並列性が現行のコンピュータシステムで使用されているやり方のただ1つの例である。

【0007】 全体の性能増加を許容するためは、多くの操作が可能な限り平行又は共起して動作できることが望ましい。この共起操作が過去に行われた1つの方法は、プロセッサからの単一サイクルが配達パッファにラッチされ、準備完了がメモリ又はI/Oデバイスへの実際に完了した書込サイクル前にプロセッサに戻される書込配達の使用によってである。全データ及びアドレス値はラッチに配達され、その後サイクルは可能になった時に目標のバスで実行される。しかしながら書込配達は、より深いレベルの配達を実施されれば当然起こり得るメモリコヒーレンシ及びサイクル順序の複雑さが故に、バス毎に1レベルのような非常に単純なレベルに保持された。それ故、もし所定バスへの単一操作より以上の書込配達ができるならば、もし複雑さが簡単に解決することができるならば得ることができる多数の利益があることは理解することができる。

【0008】 PCIバスは、全体のシステム効率特にメモリシステムの効率を増やす機会を提供する。PCIバス毎に限定された読出操作の1つは、単一ラインと同様に多数のキャッシュ又はメモリラインの読出要望を指示するために使用されたメモリ読出多重サイクルとして呼ばれることである。PCIバスサイクル定義で注目されるように、このサイクルデコードすることは、データが

実際に P C I バスで必要とされる前に得ることができるように、先読み即ちパイプラインを行なうことを始めるメモリ制御器毎に機会を提供する。メモリ制御器は、その後得られたこのデータによって、プロセッサによるアクセスを許容することができ、従って全体のシステム共起をさらに増加させる。しかしながら、P C I バスマスタがサイクルの完了前にサイクルを中止することも珍しくなく、もしそのような中止されたサイクルがメモリ読出多重サイクルを始めた後にまもなく発生するならば、先読み操作は、始められ適宜完了して、その後すぐに廃棄されるデータのみを持つことになる。これは、フルな先読み操作を始め完了する必要がない浪費操作の故に全体のシステム効率を減少させる。それ故、サイクルが P C I バス上のバスマスタによって早く中止される時には、メモリ読出多重サイクル間の先読み実行時の浪費時間を減少させることが望ましい。

【0009】更に、パーソナル・コンピュータシステムは大量生産品になって、それ故、ユーザの広範囲に変化した特定の目的に対応するために非常に柔軟性に富む必要がある。例えば、何人かのユーザはコストを度外視して究極の性能を望んでもよいが、他のユーザが非常にコストに敏感であるかもしれない。コストが性能に直接（衝突する）響く1つの領域は、主メモリに使用されるメモリデバイスの速度にある。このような衝突の他の領域は、多くの異なるマイクロプロセッサ構築及び速度毎に単一メモリ制御チップを使うことによって得ることができる規模の経済性である。しかし、単一メモリ制御器の使用は通常性能との取引きを伴う。米国特許第5, 333, 293に開示されたメモリ制御器は、多重速度のプロセッサの点を指摘したが、単一速度のメモリデバイスしか使うことができず、従ってユーザオプションを制限し、性能と取引きをしている。1993年3月22日に出願された連番08/034, 290号に開示された他のメモリ制御器において、メモリ制御器は、バンク毎又はモジュール毎の単位で異なる速度のメモリデバイスを扱うことができ、各特定のメモリデバイス毎に最適のタイミングをまだ許容する。しかしながらこのメモリ制御器は、単一速度で動作する単一プロセッサと一緒に使用する仕様であり、従ってユーザの柔軟性を提供するが、規模の経済性を提供しない。単一メモリ制御器と用いられるべき多数の形式及び速度のプロセッサを許容し、相当の複雑さを必要としないで、多数の形式及び速度のメモリデバイスの使用を許容することがより望まれる。

【0010】本発明によるコンピュータシステムは、特に P C I バス環境で多数の性能増加を提供し、多数の形式及び速度のプロセッサ及び異なる速度のメモリデバイスと容易に動作することができるメモリ制御器を持っている。

【0011】メモリ制御器は、深い書込配達を許容する

ために、プロセッサと P C I バスとメモリシステムとの間に一連の行列（キュー）を形成する。好ましい実施例において、4列の4ワードアドレスはプロセッサから配達され、書込操作毎に8列の4ワードアドレスが P C I バス配達することができる。メモリコヒーレンシは、2つの異なるウェイに維持される。どんな読出操作前にも P C I バスから受け入れられ、配達行列の両方が空でなければならない。このように全書込は、主メモリが P C I バスからの読出操作毎に一貫性があるように、読出の発生前に完了される。しかしながらより高性能がプロセッサから望まれ、それ故、コヒーレンシ（一貫性）を維持するために、連想メモリ（CAM）が P C I 対メモリ列として用いられる。プロセッサが読出要求を行なう時に、CAMは、P C I 対メモリ行列内の待機中書込操作の1つがプロセッサのその読出操作と同じアドレスにあるかどうかを決めるためにチェックされる。もしそうならば、P C I 対メモリ行列がその入口をクリアするまで、読出操作は、実行されない。もしどんなアドレスヒットも発生しないならば、読出操作は受け入れられて、仲介優先権規則によって実行される。再度このように主メモリは、発生している読出操作より前に一貫性がある。同じアドレスへの2つの書込操作を2つの行列に同席できることは、問題でなくて、バスの間の正確なタイミングがとにかく決してクリアされないように、首尾一貫した結果を発生しないことは注目される。

【0012】好ましい実施例において、メモリ読出多重が要求された時に、先読み操作の P C I バス能力が存在する。これは、メモリシステムが高い率でデータを得て、特定のサイクルで指示された時の P C I バスマスタによる読出用に、それを配達されるままにしておくのを許容する。しかしながら背景技術に注目されるように、P C I バスマスタがサイクルをその完了前に中止することは可能である。この問題を解決するために、好ましい実施例によるメモリ制御器は、P C I バスインタフェースからアボート信号を受信し、その後受信した途端たとえサイクルが十分に完了しなかった先読みであっても、D R A M データの完全性を維持しながら、先読みサイクルを終わらせる。従って先読みサイクル可能になるやいなや中止される。それ故、フルな読出は、通常の場合と同様に先読み操作間に発生するアボートの状況が性能を過度に妨げないように発生しない。

【0013】システムをさらに改善するために、好ましい実施例のメモリ制御器が、D R A M デバイスをプリチャージする時を決めるための予測規則を改善した。この予測規則は、サイクルがプロセッサから来ているか又は P C I バスから来ているかどうかに基づく。これらの新しい規則の使用によって、より効果的なプリチャージが行なわれ、追加的に、多くのページモードサイクルは先行技術によって行なわれたものと違った風に行うことができる。

【0014】最終的に、好ましい実施例のメモリ制御器は、多重速度及び形式のプロセッサ及び種々の速度のメモリデバイス毎に高度にプログラムできて、それなのに、単純にプログラムすることができる。メモリ制御器は、アドレスセットアップ（確立）及びホールド時間と、CAS*信号のパルス幅と、プリチャージ時間と、データ確立時間のような従来のDRAMサイクルの特定部分毎のクロック期間の数を指定する複数のレジスタを含む。従ってこれらのレジスタは、デザイナーが特定のプロセッサ形式及びクロック速度及びメモリデバイス速度のみを知る必要性があり、その後レジスタが小さいマトリックス又は表から適切に単純にプログラムされるように、DRAMデバイスの正常なタイミング関連及びパラメータと一致させている。複雑な操作及び置換は必要でなく、クロック期間の値が容易に決定することができる。多重プロセッサの形式及び速度及び複数の速度用の単一メモリ制御器の使用によって、規模の経済性は、ユーザの柔軟性をいまだに許容しながら、今製造量の増加によって得ることができる。

【0015】好ましい実施例の次の詳細な記述が添付図面と関連して考慮された時に、本発明のより良い理解を得ることができる。

【0016】

【実施例】次の開示は、総てが本発明の譲受人に譲渡され、参照によって本願と協働している。1994年10月14日に出願されたアランL. ゴットラム、ジェンズK. ラムゼイ及びマイケルJ. コリンズ氏による米国特許出願連番08/324, 016の表題『単一バンク、多重ウェイのキャッシュメモリ』、1994年10月14日に出願されたマイケルJ. コリンズ、ガリーW. トーメ、マイケル、モリアーティ、ジェンズK. ラムゼイ及びジョンE. ラーソン氏による米国特許出願連番08/324, 246の表題『プロセッサ及びI/Oバス操作の書込配達キューを有するメモリ制御器及びキューを制御するための順序ロジック』、1994年10月14日に出願されたウィリアムJ. ウォーカ及びアランL. ゴットラム氏による米国特許出願連番08/323, 263の表題『データエラー検知及び訂正システム』、1994年10月14日に出願されたジェンズK. ラムゼイ氏による米国特許出願連番08/324, 020の表題『もし書込が主メモリの書込保護領域に発生するならば、キャッシュメモリの部分を無効にする回路』、1994年10月14日に出願されたジェンズK. ラムゼイ及びジェフリC. スティブンス氏による米国特許出願連番08/323, 110の表題『特殊バスサイクルにตอบสนองしてキャッシュメモリを低電力モードに移行させる回路』及び1994年10月14日に出願されたジョンE. ラーソン、マイケル、モリアーティ、マイケルJ. コリンズ及びガリーW. トーメ氏による米国出願連番08/324, 011の表題『メモリへのアク

セスを仲介するためのシステム』。

【0017】今図1を参照すると、本発明によるコンピュータシステムのシステム基板Sが示される。好ましい実施例において、システム基板は、交換できる回路基板を受け入れる回路及びスロットを含む。好ましい実施例において、システム基板Sに配置された2つの1次バスがある。第1バスは、アドレス/データ部分100と、制御及びバイトイネーブル部分102と、制御信号部分104とを含むPCI即ち（周辺機器相互接続）バス98である。システム基板S上の第2の1次バスは、EISAバスEである。EISAバスEは、LAアドレス部分106及びSAアドレス部分108及びSDデータ部分110及びEISA/IISA制御信号部分112を含む。PCI及びEISAバスP及びEはシステム基板Sの骨幹を形成する。

【0018】CPUコネクタ114は、図2に示されたそのように交換できるプロセッサカードを受け入れるために、PCIバス98に接続される。PCI画像コネクタ116は、図3に示されるように、ビデオの画像カードを受け入れるために、PCIバス98に接続される。PCIオプション・コネクタ118は、PCI標準規格にデザインされたどんな追加のカードも受け入れるために、PCIバス98にも接続される。追加的に、SCSI及びネットワークインタフェース（NIC）制御器120は、PCIバス98に接続される。好ましくは、この制御器120は単一集積回路であり、PCIバスマスタ及びスレーブとして作用するに必要な能力と、SCSI制御器及びイーサネットインタフェースとして作用する回路とを含む。SCSIコネクタ122は、ハードディスク・ドライブ及びCD-ROMドライブのように種々のSCSIデバイスの接続を許容するために、制御器120に接続される。イーサネットコネクタ124はシステム基板Sに形成されて、制御器120に順に接続されるフィルタ及び変圧器回路126に接続される。これは、ローカル領域ネットワーク（LAN）にシステム基板及びコンピュータを結線するためのネットワーク又はイーサネット接続を形成する。

【0019】PCI-EISAブリッジ130は、PCIバス98及びEISAバスE間で信号を変換するため形成される。PCI-EISAブリッジ130は、必要なアドレス及びデータバッファ及びラッチと、PCIバス用の仲介及びバスマスタ制御ロジックと、EISA仲介回路と、EISAシステムとして適宜使用されたEISAバス制御器と、DMA制御器とを含む。好ましくはPCI-EISAブリッジ130は、単一集積回路であるが、他の組合せが可能である。雑多なシステムロジックチップ132は、EISAバスEに接続される。雑多なシステムロジックチップ132は、他の雑多な回路と同様に、デジタル式音声インタフェース、パーソナル・コンピュータシステムに適宜あるようなカウンタ及び

タイマ、PCI及びEISAバスP及びE用の割り込み制御器及び電力管理ロジックを含む。EISAバスEには、ISA及びEISAアダプターカードを受け入れる一連の4つのEISAスロット134が接続される。EISAバスEには統合I/Oチップ136も接続される。この統合I/Oチップ136は、好ましくはフロッピーディスク制御器、実時間クロック(RTC)/CMOSメモリ、2つのUART及び及び並列ポート及び種々のアドレス・デコード・ロジックを含む。フロッピーディスクドライブへの電線を受け入れるフロッピーディスクコネクタ138は、統合I/Oチップ136に接続される。1対の直列ポートコネクタは、並列ポートコネクタ142と同様に、統合I/Oチップ136に接続される。EISAバスE及び統合I/Oチップ136両者には、EISAバスE及びハードディスク・ドライブコネクタ146間のバッファとして作用するバッファ144が接続されて、IDE型ハードディスクドライブの接続を許容する。非揮発性ランダムアクセスメモリ(NVRAM)148は、EISAバスEに接続されて、統合I/Oチップ136からその制御信号を受信する。アドレスラッチ150は、EISAバスEに接続され、統合I/Oチップ136によって制御されて、NVRAM148用の追加のアドレス能力を形成する。好ましくはNVRAM148は、あるシステム情報を含むために使用される。データバッファ152は、コンピュータシステムの種々の追加の部品のために追加のデータ・バスXDを形成するために、EISAバスEのSD部分に接続される。NVRAM148は、そのデータビットを受信するために、XDデータ・バスに接続される。フラッシュROM154は、その制御及びアドレス信号をEISAバスEから受信して、データ転送毎にXDバスに接続される。フラッシュROM154は、好ましくはコンピュータシステムのためにBIOS情報を含んで、BIOSの改訂を考慮に入れるために再プログラムすることができる。8742即ちキーボード制御器156はXDバス、EISAアドレス及び制御部分108及び112に接続される。キーボード制御器156は、従来の設計を持ち、順にキーボードコネクタ158及びマウス或いは指針即ちポインティングデバイスコネクタ160に接続される。

【0020】また、好ましい実施例のコンピュータシステムは音声の能力を含む。この目的のため、CODECチップ162は、雑多なシステムロジックチップ132に接続され、アナログ増幅器及びミキサチップ164に接続される。FMシンセサイザ・チップ166は、アナログ増幅器及びミキサチップ164に接続され、XDバスからデジタル式の情報を受信する。また、FMシンセサイザ166は、EISAバスEの制御及びデータ部分110及び112に接続され、雑多なシステムロジックチップ132によって制御される。音声コネクタ168

は外部音声コンピュータに接続できるように形成されて、アナログ増幅器及びミキサ164の入出力に接続される。

【0021】今図2を参照するとプロセッサ基板Pは示される。図2のプロセッサ基板Pにおいて、CPU即ちプロセッサ200は、486DX/33、486DX2/66、486DX4/50~100、486DX4/33~100、486DX4/33~83、P24T、ペンティアム50/75、ペンティアム60/90及びペンティアム66/100のような複数のプロセッサ及び他の類似及び互換性プロセッサのいずれかでもあることができる。プロセッサ200は、データ、アドレス及び制御部分202、204及び206を形成してプロセッサバスPBを形成する。このプロセッサバスPBには、コンピュータシステムの性能を改善するために追加のキャッシュ能力を形成するレベル2(L2)即ち外部のキャッシュメモリシステム208が接続されている。このL2キャッシュ208は、486族プロセッサと一緒に使用された時に128キロバイト直接マップキャッシュ或いは256キロバイト2ウェイ・セット・アソシアティブキャッシュとして組織でき、ペンティアム族プロセッサと一緒に使用された時に256又は512キロバイト直接マップ又は2ウェイ・セット・アソシアティブキャッシュとして組織できるキャッシュ及びメモリ制御器(CMC)及びPCIブリッジチップ210は制御部分206及びアドレス部分204に接続されている。CMC210は、キャッシュ制御器と協働するように、L2キャッシュ208に接続されていて、それ故、L2キャッシュ208のキャッシュメモリデバイスの操作を制御する。また、CMC210は、一連のアドレス及びデータバッファ212を制御するように接続されている。データバッファ212は、主メモリ配列214へのメモリデータを扱うために用いられる。データバッファ212はプロセッサデータ部分202に接続されて、CMC210から制御信号を受信する。データバッファ212は、メモリ配列214へのメモリアドレス・バス216及びメモリデータ・バス218を供給する。メモリ制御信号バス220とメモリアドレス・バス216はCMC210から供給される。クロック分配及び発生回路222はプロセッサカードPと協働してCMC210に接続される。基板の縁のプロセッサコネクタ224は、プロセッサコネクタ114に一致するように差し込まれるように形成される。プロセッサコネクタ224は、CMC210、データバッファ212及びクロック分配回路222に接続されて、コンピュータシステムにクロックを供給し、プロセッサ200がPCI及びEISAバスP及びEにアクセスでき、PCI及びEISAバスマスタが主メモリ配列214にアクセスできるPCIインタフェースを形成する。

【0022】今図3を参照すると、実例のビデオグラフ

ックアダプタは示される。ビデオグラフィック制御器300は、グラフィックコネクタ116が一致するように差込まれるPCIグラフィックコネクタ316に接続される。ROM302はグラフィックコネクタ316に接続されて、ビデオグラフィック制御器300から制御信号を受信する。ビデオメモリ304は、グラフィックデータを記憶するために使用され、ビデオグラフィック制御器300及びデジタル/アナログコンバータ(DAC)306に接続される。ビデオグラフィック制御器300は、ビデオメモリ304の操作を制御して、データが書込まれ、必要時に回復するのを許容する。ビデオのコネクタ308は、DAC306に接続される。モニタ(図示略)はビデオのコネクタ308に接続される。

【0023】上述のコンピュータシステムは完全さのために提供され、多数の変形例が当該技術者に明らかに開発することができることが注目される。

【0024】今図4を参照すると、メモリ制御器210のブロック図が示される。メモリ制御器210における3つの主要制御ブロック及び3つの一次アドレス転送行列がある。3つの第1の制御ブロックは、プロセッサ制御即ちPCONブロック400、PCI制御即ちICONブロック402及びメモリ制御即ちMCONブロック404である。PCONブロック400は、プロセッサバスPB特にプロセッサアドレスバス204及びプロセッサ制御バス206へのインタフェースを形成する。更に、PCONブロック400はL2データキャッシュ208に接続されて、必要な制御信号を供給する。L2キャッシュ制御器は、PCONブロック400に含まれる。追加的にPCON400は、データバッファ212を制御する信号を供給する。ICONブロック402は、データバッファ制御信号をデータバッファ212に供給し、それに加えて、特に、制御及びアドレス部分をPCIバス98にインタフェースする。MCONブロック404は、メモリデバイス制御部分であって、メモリアドレス・バス216及びメモリ制御バス220に接続されて、更に、データバッファ212を制御する信号を供給する。各特定の制御ブロック400、402及び404は、説明されるように、データバッファ212の異なる部分を制御する。

【0025】メモリ制御器210及びデータバッファ212は、データ転送がプロセッサバスPB及びメモリバスの間、PCIバス98及びメモリバスの間又はプロセッサバスPB及びPCIバス98の間で発生することができるように、三角形又はデルタ結線のように効果的に組織される。この目的のためにPCONブロック400はICONブロック402に直接接続されて、PCIバス98へのプロセッサ200による読出操作のために読出アドレスを供給する。書込アドレスは、PCON400からP2I行列406に供給される。好ましくはP2I行列は、4つの書込操作が行列に配達することがで

き、従って、一度に待機中であることができるように、4段階の深さの操作である。64ビットペンティアムマイクロプロセッサ200の場合において、これが8個の32ビットPCIバス98操作に翻訳されることが注目される。P2I行列406の出力がICONブロック402に供給されて、アドレスの流れを形成する。類似ファクションにおいて、読出アドレス情報及びアドレスバンク情報は、主メモリ214からのプロセッサ読出のためにPCONブロック400からMCONブロック404に供給される。書込アドレス及びバンク情報は、PCONブロック400から、再度4段階の4ワードアドレス深さが好ましいP2M行列408に供給されて、もしペンティアムプロセッサならばプロセッサ200からの4つの書込操作、もし486基準プロセッサならば8つの書込操作がメモリ配列214に配達することができる。P2M行列408の出力が、MCONブロック404に供給される。

【0026】ICONブロック402はMCONブロック404に接続されて、PCIバス98からメモリ214への読出操作毎に読出アドレス及びメモリバンク情報を形成する。それに加えて、書込アドレス情報及びバンク情報はICONブロック402からI2M行列410に供給される。好ましくはI2M行列410は、PCIバス98からメモリ214へのかなり深い配達を形成するために、8段階の4ワードアドレスを配達することができる。I2M行列410の出力はMCONブロック404に接続される。SMAP即ちシステムマップブロック412は制御ブロック400、402及び404に接続されて、必要な種々の信号を供給する。

【0027】それに加えて、M2IABORTとして参照された信号はICONブロックから402MCONブロック404に供給されて、メモリ読出多重操作が他のPCI中止操作と同様にPCIバスマスタによって中止された時をMCONブロック404が決定できるように許容する。この議論のためにメモリ読出多重の場合のみに関心があり、他のケースは説明しない。MCONブロック404は、スヌープ要求即ちSNPREQ信号、メモリ対PCI読出操作作用のアドレスであるM2I読出アドレス及びPCONブロック400へのI2M書込アドレスを形成する。これは、PCONブロック400がL2キャッシュ制御器とのスヌープ操作を実行し、プロセッサ200への操作を形成することを許容して、プロセッサ200の内側のL1キャッシュ制御器もスヌープ操作を行なうことができる。書込アドレスと同様に読出アドレスのスヌープは、L2キャッシュ制御器及びある場合のプロセッサ200内のL1キャッシュ制御器がライトバックキャッシュ制御器として好ましく組織され、それ故、スヌープ操作がメモリコヒーレンスを維持するために読出で発生しなければならないので必要である。PCONブロック400は、SNPDONE即ちスヌープ

実行済及びSNPHITM即ちスヌープヒットをMCONブロック404への修正信号に形成して、MCONブロック404が読出又は書込操作もし適切であるならば読出操作の再トライを進行できるようにする。

【0028】今図5を参照すると、データバッファ212の多くの詳細が示される。データバッファ212もメモリ制御器210のそれらと同様に、行列411、413及び414を含んで、データが維持される間にアドレスがメモリ制御器210で追跡されて、データバッファ212を通して転送されることが注目できる。メモリ制御器210は、特定の制御ブロック400、402及び404がデータバッファ212の各部分を制御するようにデータバッファ212を制御するために組織される。例えば、PCONブロック400は、プロセッサデータバスから種々のレジスタへのデータのラッチ及びメモリからプロセッサデータバス202へのデータの出カイネーブル即ち駆動を制御する。同様にICONブロック402は、PCIバス98から種々のレジスタへのデータのラッチ及びPCIバス98へのデータの出カイネーブルを扱う。MCONブロック404は、種々の行列411、413及び414を通してデータをシフトし、行列411、413及び414から実際のメモリデバイスへのデータを駆動し、プロセッサデータバス202及びPCIバス98に供給すると同様に、メモリデバイスからI2M行列414又はレジスタに必要とされるデータをラッチする操作を扱う。プロセッサ対メモリ及びプロセッサ対PCI行列411及び413はデータバッファ212に単一方向であるが、PCI対メモリ行列414が両書込データ及び先読みデータ用に使用されることを双方向性的に操作されることが注目される。操作は下記に従ってクリアされる。

【0029】好ましい実施例のメモリ制御器210の特徴の1つは、種々の速度のマイクロプロセッサ及びメモリ配列214における種々の速度のDRAMデバイスの使用に無制限に近い柔軟性を許容することである。実にそれは、メモリデバイスがバンクからバンクに変化でき、好ましくは主メモリ配列214が8バンク又はモジュールから形成される。好ましい実施例のメモリ制御器210は3種類のメモリ速度を用いることができ、各バンクが特定の速度の1つとして指定される。その後各特定のメモリ速度は、DRAM仕様書に一般に用いられるように、タイミング値に関連する一連のパラメータを持っている。各特定のメモリ速度毎に、CAS*ストロブへのメモリデータ確立時間及びCAS*ストロブのパルス幅と同様に、RAS*からのローアドレスホールド時間を指示する値が形成され、読出及び書込毎のCAS*ストロブに対するカラムアドレス確立時間が形成され、RAS*信号に対するローアドレス確立時間が形成される。それに加えて、RASプリチャージ時間は、各メモリ速度毎に形成される。サイクル準備完了信号が

プロセッサ200又はPCIバス98に供給される時の指示もタイミングパラメータとして示される。好ましくはこれらのタイミングパラメータは、構成が表1に示された組織16ビットレジスタのメモリの10ビットを用いている。

【0030】表 1

メモリタイミング制御レジスタ

ビット15予約

ビット[14:13] RASプリチャージ

00=2クロック

01=3クロック

10=4クロック

11=5クロック

ビット[12:11] MRDY読出状態

00=C1状態でMRDYアサート

01=C2状態でMRDYアサート

10=C3状態でMRDYアサート

11=予約

ビット[10:9] 予約

20 ビット8、CASパルス幅

0=2クロック

1=3クロック

ビット[7:5] 予約

ビット4、CASへのMD確立

0=1クロック確立時間

1=2クロック確立時間

ビット3、RASへのローアドレス確立

0=1クロック確立時間

1=2クロック確立時間

30 ビット2、CASへのカラムアドレス書込確立

0=1クロック確立時間

1=2クロック確立時間

ビット1、CASへのカラムアドレス読出確立

0=1クロック確立時間

1=2クロック確立時間

ビット0、RASからのローアドレスホールド

0=1クロックホールド時間

1=2クロックホールド時間

40 【0031】RASプリチャージは、全てのRAS*信号が再びローにアサートできる前にハイにネゲートされなければならないクロック数を定義する。MRDY読出状態は、どのCAS状態MRDYがメモリデータ読出毎にアサートされるべきであるかを定義する。もしCASパルス幅が2クロック幅であるならば、C2はスキップされる。MRDY状態は、クロック周波数及びメモリ速度で異なる。CASパルス幅は、どの数のクロックのCAS*信号がアサートされ続けられなければならないかを定義する。CASへのMD確立時間は、CAS*が書込用にアサートされる前に、MDバスがどの数のクロックでデータバッファ212から駆動されなければならない

15

いかを定義する。CASへのカラムアドレス書込確立は、CAS*が書込サイクル用にアサートできる前に、カラムアドレスが発生された後のクロック数を定義する。CASへのカラムアドレス読出確立は、CAS*が読出サイクル用にアサートできる前に、カラムアドレスが発生された後のクロック数を定義する。RASからのローアドレスホールドは、RAS*がアサートされた後でローアドレスが変化する前のクロック数を定義する。

【0032】その後図6を参照すると、各記憶バンクは、特定の記憶バンクの最後のアドレスを示す関連したレジスタ500、502及び504を含む。これらのアドレスは、特定のバンクに配置されたメモリモジュールの特定のサイズを決定した後にプロセッサ200によって書込むことができる。その後バンク終端アドレスレジスタ500、502及び504の出力は、比較器506、508及び510に1つの入力として供給される。比較器506、508及び510への第2入力がプロセッサバスPB又は適切なPCIバス98のような特定のモジュールのメモリアドレスによって供給される。8つのバンクの3つのみが単純さのために図示されることは注目される。比較器506、508及び510には2つの出力と第3の入力を持つ。1つの出力は、もし供給されたメモリアドレスが特定のバンクの最後のアドレスより小さく、比較器のイネーブルを示す値にキャリーがあるならば、その値より少ないものが真即ちハイで、これが選択されたバンクであることを指示するように、出力より小さい。比較器の第2の出力は、次の比較器をイネーブルするために使用された信号より大きいとか等しい。例えば、比較器506は、比較器508のイネーブル入力に接続された出力以上のものを持って、もしメモリアドレスがバンク0の終端アドレスレジスタ500に形成されたアドレス以上であるならば、比較器508が活性化される。その後もしアドレスがレジスタ502に記憶されるようにバンク1の最後のアドレスより小さいならば、比較器508のBANK1信号の出力が活性化される。その代わりにもしメモリアドレスがより大きいならば、この連鎖は、メモリアドレスがバンクのいずれかに含まれるもの以下であるかどうかを決めることを続ける。それ故、特定の記憶バンク中のメモリアドレスの存在を示す8つの特定のバンク出力(BANK0~BANK7)がある。

16

【0033】これらのBANK0~BANK7信号は、MCONブロック404への準備のために、12M行列410又はP2M行列408中の書込アドレス値に沿って記憶されたBANK<2:0>信号を出力する記憶バンク値エンコーダ512に供給される。

【0034】上記注目されるように、各バンクは、3つの特定のメモリ速度の1つをとることができ、これらの指示は、レジスタ514及び516に含まれる。2つのレジスタのみが単純さのために示している。レジスタ514及び516の出力は、特定の速度を示す2ビットであり、これらのビットは、選択入力BANK<2:0>信号に接続される8入力2ビットマルチプレクサ518の入力に供給される。BANK<2:0>信号には特定のアドレスが供給されて、マルチプレクサ518の出力がMCONブロック404がメモリ操作を実行しようとしている特定のアドレス用のバンクを示す特定の2ビットであるように、PCONブロック400又はICONブロック402用のアドレスをマルチプレクサ518に読出或いは書込む。マルチプレクサ518の出力は、3入力マルチプレクサ520の選択入力に接続される。マルチプレクサ520は、速度0、1及び2毎のメモリ速度タイミング値を含むレジスタ522、524及び526にそれぞれ接続される。これらのメモリタイミング値は、表1に示されたものである。マルチプレクサ520の出力はMCONブロック404のために後述されるステートマシン及びロジックで用いられるタイミング値である。それに加えて、アドレス値<27...3>及びバイトイネーブル即ちBE*<7...0>信号も、それらが行列408又は410を通して追跡され或は読出操作で直接供給されるように、MCONブロック404に供給される。それ故、メモリバンク速度タイミング値は、まず特定バンクをデコードし、その後オプションなメモリ速度セッティングの既に定義され記憶されたメモリ速度値に基づくバンク毎に早見を持ち、その後その特定のメモリサイクル毎の操作用に特定のメモリを選択することによって、直接供給される。

【0035】3種類のプロセッサ速度及び2種類のメモリデバイス速度毎の好ましい実施例用のタイミング値の例が表2に示される。

【0036】

表 2

メモリタイミング制御合同

パラメータ	33MHz		50MHz		60MHz	
	60ns	70ns	60ns	70ns	60ns	70ns
RASプリチャージ	2	2	3	3	3	4
MRDY状態アサート	C1	C1	C3	C3	C2	C3
CASパルス幅	2	2	2	2	2	3
MD確立時間	1	1	1	1	2	2
書込カラムアドレス確立時間	1	1	1	1	1	1

17				18			
読出カラムアドレス確立時間	1	1	1	1	1		
RAホールド時間	1	1	1	2	2	1	
メモリタイミング制御	00 00	00 00	30 00	30 01	20 11	51 10	
レジスタ (16進数)							

【0037】DRAMデバイス速度は、S I M M上に適宜形成されたエンコードビットを讀出し、容量をチェックすることによって決定することができる。その後これらの2つのパラメータは、S I M M速度を決定するために、早見表で使われる。この技術は、1993年3月22日に出願された連番08/034、105号に十分に記載され、これによって参照物として協働する。それで各バンク毎に公知なDRAMデバイス速度及び他のエンコードビットに公知なマイクロプロセッサ速度によって、これらの2つのパラメータは第2の早見表に用いられて、その連続のためにタイミング値レジスタに配置されるべき2進数値を得る。コンピュータシステムのデザイナーは、種々のバスの種々の伝播遅延を知り、その後各結合毎のタイミング値を決定するために、メモリデバイス製造業者によって提供されるクロック速度及びメモリタイミングパラメータを使うことができるのみである。

【0038】その後図7に進行して、アドレス、バイトイネーブル及びタイミング値の使用がMCONブロック404の部分に供給されるように示される。プロセッサ書込要求、プロセッサ読出要求、リフレッシュ要求、P C I 読出要求及びP C I 書込要求信号は、I 2 C A M H I T、I 2 M Q E M P T Y及びP 2 M Q E M P T Yとして参照される信号に沿うアービタ550への入力である。種々の要求信号は、示された形式の示された源から操作するメモリ制御器のために操作が待機中であることの指示である。これらの信号の開発が十分に説明しないが、当業者が容易に開発することができる。アービタ550の出力は、操作の源及び方向が発生したことを指示するP 2 M、M 2 P、I 2 M及びM 2 I 信号、即ちプロセッサ対メモリ書込用のP 2 M、プロセッサ対メモリ読出用のM 2 P、P C I 対メモリ書込操作作用のI 2 M及びP C I 読出操作作用のM 2 Iである。これらの4つの信号は、特定の読出又は書込アドレス、バイトイネーブル及びメモリタイミング値及び特定の源毎のバンク指示を受信する4入力マルチプレクサ552への選択信号として供給される。P 2 M行列408は、プロセッサ対メモリ書込操作作用の行列408から書込アドレス、バイトイネーブル、メモリタイミング及びバンク情報を形成し、一方M 2 Pの場合読出アドレス、バイトイネーブル、メモリタイミング及びバンク情報がプロセッサ読出のために直接供給される。同様にI 2 M行列410は、I 2 M行列410からの書込アドレス、バイトイネーブル、メモリタイミング及びバンク情報を形成し、一方読出アドレス情報がI CONブロック402からMCONブロック404に直接供給される。マルチプレクサ552の出力は、適切であるものとして読出か書込のための必要な

メモリ位置をアクセスするために、特定のアドレス値、バイトイネーブル、どの特定のバンクを指示するバンク値及びメモリ制御器210によって実行されるべき特定のメモリ操作毎のメモリタイミング値である。ページヒット検知器554はマルチプレクサ552の出力に接続されて、ページヒットが発生したかどうかをメモリ制御器210が決定できるPAGE HIT信号を形成して、最良の性能のために、DRAMデバイスを所望のページモード操作で操作することができることは注目される。

【0039】その後図8に進行して、MCONブロック404において用いられた種々のステートマシン及び他のロジックを指示するブロックが示される。多くのステートマシンは、実際の情報及び制御機能を形成して、組み合わせられる。第1のステートマシンは、MEMSM即ちメモリステートマシン600である。メモリステートマシン600は、マルチプレクサ552からのメモリタイミング値及びI CONブロック402からのM 2 I A B O R T信号を受信する。プリチャージステートマシン602は、もしページモード操作が発生していないならば、特定の記憶バンク用のプリチャージ時間が完了したことを指示するR A S P C H G即ちR A S プリチャージとして参照される信号を供給するために形成される。R A S P C H G信号は、メモリステートマシン600に供給される。更に、R A S U P即ちR A S アップロジック604は、メモリサイクルが待機中でない時に、R A S *即ちローアドレスストロブ信号がハイ又はローを許可するかどうかを決めるために、メモリステートマシン600によって用いられるR A S U P信号を形成し、従って次の操作が性能を改善するためにページヒット又はページミスであるかの予測を形成する。このロジックの詳細な操作が後述される。

【0040】リフレッシュステートマシン606は、リフレッシュ操作を制御するために形成されるが、その操作がかなり従来のであるので、この明細書に詳細に説明されず、明快のために省略される。メモリアドレスステートマシン608は、システムの基本クロック信号と同様に、マルチプレクサ552からタイミング値を受信して、メモリステートマシン600への連結信号、M S E L R A即ちメモリ選択ローアドレス信号及びM W E *即ちメモリ書込イネーブル信号を出力する。M S E L R A信号は、選択入力に基づいてロー及びカラムアドレスを供給するために、適切に接続されたメモリアドレスをその入力を受信する2:1マルチプレクサ610の選択入力に供給される。バーストアドレスブロック612は、カラムアドレスに供給及び接続されて、バースト操作を簡単にする。マルチプレクサ610の出力は、特定

の操作毎にメモリアドレス・バス216上でメモリ配列214に形成されるメモリアドレスである。同様にMWE*信号は主メモリ配列214に供給されるようにメモリ書き込みイネーブル信号である。メモリデータ制御ステートマシン614は形成される。それは、あるメモリタイミング値を受信し、メモリステートマシン600に連結信号を出力し、データバッファ212の適切な部分の操作を制御する。

【0041】メモリステートマシン600は、MRAS即ちマスタRAS及びMCAS即ちマスタCAS信号を供給する。MRAS信号は特定のメモリ操作のためのバンク値信号と結合して、特定のバンクに供給されたRAS* $<7:0>$ 信号を形成する。もしこれが操作のための特定のバンク値であることをバンク値が指示するならば、RASロジック616は、反転形式でその特定のバンクにMRAS信号を出力する。もしバンク値が特にアドレスされたものでないならば、その特定のバンクのためのRAS*信号がハイに保持されて、メモリデバイスが十分にプリチャージされるのを許容する。MCAS信号は、好ましい64ビット幅用のBE* $<7:0>$ 信号を受信するCASロジック618にも供給される。これらの信号は、RAS* $<7:0>$ 信号によって指示されたバンクの特定のメモリデバイスをイネーブルするために使用されたCAS* $<7:0>$ 信号を生成するために、MCAS信号とエンコードされる情報の8バイトレーンを提供する。もし特定のバイトイネーブルが活性化されないならば、その関連したCAS*信号が活性化されず、それ故、メモリデバイスは、カラムアドレスを受信しないように選択されない。これは、バイトレーンエンコードを形成するCAS*信号によって単一MWE*信号の使用を許容する。

【0042】行列406、408及び410は適切な幅の一連のレジスタとして組織される。各行列における特定のレジスタは、各行列と協働する3つの特定のカウンタ値に基づいて選択される。3つのカウンタは、読出位置カウンタと書込位置カウンタとデータ内容カウンタである。読出位置カウンタは、次のサイクル用の読出データが得られるべきで特定ブロックに供給される4又は8つのレジスタの特定の位置を決定するために用いられる。書込位置カウンタは、データが次の操作で書かれるべき特定のレジスタを指示するために用いられる。読出及び書込位置カウンタは、各読出及び書込操作でそれぞれインクリメントされて、従って、循環ファッショで操作される。データ内容カウンタは、行列に実際にデータがあるか否かを指示するために用いられる。データ内容カウンタは、アップダウンカウンタである。データ内容カウンタは行列への書込操作で計数が増え読出操作で減少する。もし行列が最大値にあることによってフルであることをデータ内容カウンタが指示するならば、データが行列から除去されるまで、データは行列に書込ま

ていない。これらのカウンタは従来のロジックであって、単純さと明快さのために示さない。従ってI2MQ EMPTYとP2MQ EMPTY信号は0値でのデータ内容カウンタに対応する。

【0043】I2M行列410の長さの故に、連想メモリ即ちCAMとして組織される。I2M行列レジスタ640及び642は図9に示され、多くが単純さのために除去される。特定のレジスタ640、642の出力は、マルチプレクサを経由してMCONブロック404にのみ供給されるが、協働の比較器644及び646にも供給される。比較器644及び646の第2入力がプロセッサアドレスバス204に供給されたプロセッサアドレスを受信する。比較器644及び646は、もしプロセッサアドレスバス204で直接供給されたアドレスがI2M行列410に含まれるアドレスの1つと等しいならば、特定の比較器は、その等しい信号を真に形成するように、等しい出力を持っている。等しい信号の出力はANDゲート648及び650の一入力に行く。ANDゲート640及び650への第2入力が有効ビットレジスタ652及び654によって供給される。もしデータがI2M行列410の特定のレジスタ640、642に含まれないならば、有効ビットレジスタが0にセットされ、もしデータがあるならば、ビットは1にセットされる。これらの有効ビットレジスタは、データがI2M行列410に書込まれる時にセットされ、データが行列410から読出される時にクリアされる。この様に、もしレジスタが有効で比較器が等しいことを決定するならば、特定のANDゲートの出力が、ヒットが発生したことを指示するために1つの値にセットされる。ANDゲート648～650の出力は、プロセッサによって要求される特定の読出アドレスがPCIバス98からの書込操作としてI2M行列410に実際に存在していることを指示するために形成されるI2MCAMHIT信号を出力する8入力ORゲート654に供給され、それ故プロセッサ200からのメモリ読出要求は、特定の位置がI2M行列410からフラッシュされるまで、発生することができない。この遅延の操作が後述されるようにアービタ550で見られる。

【0044】好ましい実施例において、主メモリ214用の5つの可能な要求即ちプロセッサ対メモリ書込(P2M)、メモリ対プロセッサ読出(P)、PCI対メモリ書込(I2M)、メモリ対PCI読出(M2I)及びリフレッシュがある。P2M書込はメモリ214への書込を参照し、M2P読出がメモリ214の読出を参照し、両者がマイクロプロセッサ200によって始められる。I2M書込はメモリ214への書込を参照し、M2I読出がメモリ214の読出を参照し、両者がPCIバスマスタによって始められる。EISAバスマスタからの全てのメモリ要求は、PCIバス98上のバスマスタであるために必要なロジックを含むPCI-EISAブ

リッジ 130 に通される。従ってどんな EISA 起源のメモリ要求も PCI-EISA ブリッジ 130 によって有効にアサートされたメモリ要求である。一般にメモリ要求の優先権は、幾つかの例外を除いて次の (1) 第 2 リフレッシュ要求、(2) P2M 書込要求、(3) M2P 読出要求、(4) I2M 書込要求、(5) M2I 読出要求、及び (6) 第 1 リフレッシュ要求である。第 2 リフレッシュ要求は 2 つのリフレッシュが未解決であることを指示する。これが発生した時には、メモリ制御器 234 が未解決の両リフレッシュ要求を最も高い優先権に設定して、両リフレッシュサイクルを実行する。P2M 書込要求は、優先権が第 2 リフレッシュを除いて他のメモリ要求よりいつも高い。しかしながら、もし I2M 行列 410 がフルであり或いは I2M 行列 410 が空からでなく、PCI バス 98 読出が未解決であるならば、信号 M P2M NOPOST はアサートされて、PCI 書込操作が完了するまで、P2M 書込要求の更なる列に並ばせることを防止する。これは、P2M 行列をクリアできて、それによって PCI バス 98 からの要求を点検するのを許容する。しかしながら信号 M P2M NOPOST のアサートは、ライトバックが PCI メモリ要求で必要であり得るので、ライトバックサイクルを列に並ばせるのを防止しない。

【0045】M2P 読出要求は、優先権が P2M 書込要求よりいつも低い、通常優先権が I2M 書込及び M2I 読出要求より高い。操作は、P2M 行列 408 が空である時にのみ発生することができる。しかしながら、未ロックされた M2P 読出要求は、もし M2P 読出が I2M 行列に待機中の I2M 書込と同じアドレスにあるならば、優先権が I2M 書込要求より低く強制される。これが発生した時に、M2P 要求は、一致したアドレスを持つ I2M 書込要求が主メモリ 214 に書込まれるまで、優先権が I2M 要求より低いままである。また、M2P 読出要求は、もし I2M 行列がフルであるならば、優先権が I2M 書込要求より低く強制される。追加的に、もし I2M 書込要求が待機中である間に M2I 読出要求がアサートされるならば、I2M 書込要求は、優先権が M2P 読出要求より高く強制されて、I2M 行列がクリアでき、それによって M2I 要求を進行することができる。更に、M2I 読出要求は、もし M2I 読出が M2P 要求毎に待機して 1 つ以上の仲介サイクル毎にネグートするならば、優先権が M2P 読出要求より高く強制される。

【0046】I2M 書込要求は、優先権が常に第 2 リフレッシュ要求及び P2M 書込要求より低く、上記例外を除いて優先権が一般に M2P 読出要求より低い。I2M 書込操作は、P2M 行列 408 が空である時にのみ発生することができる。I2M 書込要求は、優先権が常に M2I 読出要求より高い。I2M 書込要求は、もしプロセッサが主メモリ 214 のロックされたアクセスを行なっ

ているならば遅延される。従ってロックされたプロセッササイクル毎に、上記説明の例外は、M2P 読出要求のより高い優先権を I2M 又は M2I 要求で無効にすることを適用しない。

【0047】主メモリ 214 のロック即ち原子力のアクセスは、マイクロプロセッサ 200 で駆動される信号 LOCK* によって示される。ロックされたサイクルは、マイクロプロセッサ 200 が主メモリ 214 のアドレス位置を讀出すことができ、マイクロプロセッサ 200 が同じメモリ位置に戻って書込む前に、アクセスされた位置が他のバスマスタによって変更されないことを保証する。これらの型のサイクルが読出修正書込サイクルとして参照される。また、ロックされたサイクルは、そのオペランドの 1 つがメモリ基準である時の XCHG (交換) 命令の実行中、セグメント (区分) 又はページテーブル入口 (エントリ) を更新している時及び割込みアクトリッジサイクルを実行している時のような他のバス転送中に発生される。

【0048】M2I 読出要求は、常に優先権が第 2 リフレッシュ要求、P2M 書込要求及び I2M 書込要求より低い。しかしながらそれは、優先権が上記例の未ロックされた M2P 読出要求より高い。M2I 読出操作は、P2M 行列 408 が空である時にのみ発生することができる。

【0049】最終的に第 1 リフレッシュ要求は、常に優先権が他の他の要求より低い。しかしながら上記注目されるように第 2 リフレッシュ要求がアサートされる時には、両方の第 1 及び第 2 リフレッシュ要求は、他の要求が待機中であるかどうかに関わらず実行される。

【0050】この仲介の更なる詳細が、これと同時出願された連番 08/324, 011 の表題『メモリへのアクセスを仲介するシステム』で提供され、これによって文献として協働する。

【0051】今図 10 を参照すると、メモリステートマシン 600 が詳細に示される。特定の状態バブルの次のイタリック体の表示は、ステートマシンによって展開せられた種々の出力信号、この場合 MRAS 及び MCAS 信号の値を指示する。1 つのバブルから他のバブルに行く矢印に隣接したフレーズは、そのバスが受け入れた論理的条件である。もしバスが隣接した論理的条件を持たないならば、そのバスは他の全ての場合で受け入れられる。移行はプロセッサクロック信号の上昇縁で実行される。コンピュータのリセット時には、ステートマシン 600 の操作は、RI 即ち RAS アイドル状態に移移する。この状態において、もし次の状態が R1 状態でないならば、MRAS 信号は、アサートされない。もし次の状態が R1 であるならば、MRAS 信号は値 1 にセットされ、即ちローアドレスストロブ信号が供給されるようにアサートされる。MCAS 信号は、0 レベルにネグートされる。制御は、もしプロセッサ読出操作が発生す

るようにM2P状態が真であり、もしPCI読出操作が発生し(M2I)中止されず(!M2IABORT)クリアされず(!CLR I2M)或はもしプロセッサ又はPCIバスからの書込操作が発生して、RAS信号へのローアドレス確立時間がメモリアドレスステートマシン608によって指示されるように一致したことを指示するローアドレス準備完了信号(RADRRDY)が真であり、或はプロセッサ読出場合においてももし中止されたメモリ対PCI先読みが完了したならば適切に指示されるどんな待ち時間信号もなく(WAITRADRRDY)、かなり速いメモリがあることを指示するローアドレス確立時間2(RADSETUP2)信号が活性化されないならば、及びリフレッシュサイクル(RFSHON)が発生していず、PCIバス98への現行の読出サイクルが中止されておらず(M2IABORTCUR)、プリチャージ期間がRAS PCHG信号によって示されるように完了したならば、RI状態からR1状態に進行する。M2IABORT信号は、上述されるようにICONブロック402によって供給される。M2IABORTCUR信号は、MCONブロック404によって現在実行されているサイクルができるだけ早く中止又は終了されるべきであることを指示する。M2IABORTNEXT信号は、待機中の次のM2Iサイクルが中止されるべきで、サイクルを単にスキップすることによって発生することを指示する。CLR I2M信号の展開は、詳しく後述するが、要するに、I2M行列414をフラッシュして読出サイクルを再実行しなければならないように、修正された位置へのスヌープ読出ヒットが発生したことを指示する。それ故、もしメモリサイクルが指示され、列アドレス確立時間が一致され、デバイスがプリチャージされるならば、制御はRI状態からR1状態に進行する。他の全ての場合において制御はRI状態に留まる。

【0052】R1状態において、MRAS信号は1即ちハイ及び真にセットされ、MCAS信号はローにセットされて、これがサイクルのRASのみの部分であることを指示する。制御は、もしRADHLD2信号が真であって、ホールド時間の2クロック期間が必要とされそれ故中間のR2状態が必要とされることを指示するならば、R1状態からR2状態に進行する。もしRADHLD2信号が真でないならば、ホールド時間の1クロックのみがRAS*信号の活性化から要求されて、制御はR1状態からR3状態に直接進行する。R2状態において、MRAS信号はアサートされ、MCAS信号がネゲートされる。制御はプロセッサクロック信号の次の上昇縁でR2状態からR3状態に進行する。

【0053】R3状態において、MRAS信号は真即ちハイであり、一方MCAS信号は、もし次の状態がC1状態であるべきならばハイにセットされ、さもなければMCAS信号がローにセットされる。制御は、もしカラ

ムアドレス準備完了(CADRRDY)信号がメモリアドレスステートマシン608によって形成されるように真であって特定のラムアドレス確立時間を指示するならば、読出又は書込が満たされ、PCI読出操作が発生し或はメモリ対プロセッサバス読出操作が発生しているならば、及びメモリ操作が読出データ毎に待機して止まらないならば、或は、プロセッサ対メモリ又はPCIバス対メモリからの書込操作がメモリデータステートマシン614によって形成されるMDVALID信号によって指示されるように有効であるデータで発生するならば、R3状態からラム状態の最初のC1状態に進行する。それ故、もしラムアドレス確立時間が一致し、書込場合毎に適切に有効である書込操作データの指示が形成されるならば、制御はR3状態からC1状態に進行する。さもなければ制御は、満足すべきタイミング状態のために待機するR3状態に留まる。

【0054】C1状態において、MRAS及びMCAS信号はサイクルのCAS部分が発生していることを指示するためにハイである。制御は、もしCAS PW3即ちCASパルス幅3クロック信号が真であるならば、C1状態からC2状態に進行する。制御は、もしCAS PW3信号が真でないならば、C1状態からC3状態に直接進行し、制御が常にプロセッサクロック信号の次の上昇縁でC2状態からC3状態に進行する。C2状態において、MRAS及びMCAS信号も真である。従って、C2状態は、必要時に供給される追加のクロック期間を有するより遅いプロセッサ又はより速いメモリが真であるように、もしCASパルス幅がより狭くできるならばスキップされる。

【0055】C3状態において、MCAS信号は0レベルにネゲートされて、特定のメモリサイクルの完了を指示し、MRAS信号は、もし次の状態がRI状態であるならば或はもし特定のサイクルが中止されるならば0にセットされる。さもなければMRAS信号はハイ即ちアサートされた状態に留まって、もし適切ならばページモードの操作を許容する。制御は、もし486及びペンティアムプロセッサ用の望ましい操作であるバースト操作が発生せず或は現行のメモリ対PCI読出サイクルが後述されるように中止され、もし次のメモリ対PCI読出操作が中止されず、M2IABORT信号又はCLR I2M信号によって示されるようにM2Iサイクルが中止されるべきことのどんな指示もなく、非リフレッシュサイクルが発生し(M2P+ADRRDY)これがページヒットでなく、リフレッシュサイクルが発生し或いはどんなサイクルも待機中でなくて、次のサイクルの予測がページミスであることをRASUPブロック604が指示したことを示すRASUP信号が真であるならば、C3状態からRI状態に進行する。RASUP信号は、次に述べる通り、発生される。

【0056】RAS UP=!(!WRITEBAC

K && P2M) || (!MDATARD && M2P) || (!Q I2MQEMPTY && I2M) || (!M2IABORTCUR && M2IMULREAD && !M2IABORTNEXT) | | HIT2MOD)

【0057】それ故、RAS*信号はロー即ち、次のL1或はL2キャッシュからのライトバックではないプロセッサ対メモリ書込操作、プロセッサコード読出操作、待機中の書込操作を有するPCIバス書込操作、中止されなかったPCIバス98からのメモリ読出多重操作或いはライトバック操作が発生したのみを指示する修正されたアドレスへのヒットをページモードに保持される。他の全ての場においてRASUP信号は、プリチャージがアイドル期間中に発生できるようにフルな新規ページアクセスを始めるために良好と考慮されるように真である。

【0058】もしC3状態からRIへの遷移のための状態が真でないならば、制御はC3状態からRN状態即ちRASローアイドル状態に進行する。この状態において、MRAS信号はもし次の状態がRI状態であるならばローであり、さもなければMRAS信号中がアサートされる。MCAS信号はもし次の状態がC1であるならばアサートされ、さもなければMCAS信号がロー状態に留まる。制御は、もしカラムアドレス確立時間が一致し、発生しているどんな中止操作もないならば、概してバースト操作毎にRNからC1状態に進行する。制御は、サイクルが中止され、即ちページミスであるプロセッサ非バースト読出が次に発生することを一般に指示する状態でRN状態からRI状態に進行する。他の全ての場において制御は、サイクルを実行する必要がある或いは中止又はページミスが発生するまでローのRAS*信号でアイドルするRN状態に留まる。それ故、メモリステートマシン600は、適当なバンクのタイミング値に基づいてMRAS及びMCAS信号を駆動し、これらのタイミング値が上述されるように変化できることが理解できる。

【0059】メモリアドレスステートマシン608は図11に示される。ステートマシン608は、リセット時にRAI状態から始まる。この状態においてRADRRDY信号及びCADRRDY信号は、列及びカラムアドレスが準備できていないことを指示するためにローにセットされ、MSELRA信号は、まず、メモリデバイスに供給されるべきローアドレスを選択するためにハイにセットされる。MWE*信号は、リセット時及びRAI状態に入った後に1即ちネグートレベルにセットされ、後述されるように、CH1状態から入った第1状態後のCH1状態値と等しくされる。制御は、もしアクティブなサイクルがPCIバス98又はプロセッサ200から待機中であり、先読み操作の次のサイクルが中止されるべきでないならばRAI状態からRH1状態に進行する。

現行のM2I読出サイクルは、M2IABORT信号の受入或はライトバックが発生しているようなある状態の下で中止され、一方次のM2I読出サイクルはM2INA信号が受信された時に中止されるがM2IABORT信号が受信される時にサイクルが始まらない。他の全ての場において、制御がRAI状態でループする。

【0060】RH1状態において、CADRRDY信号がローにセットされ、RADRRDY信号は、RAI状態から入った時にM2P即ちメモリ対プロセッサ読出の値にセットされ、後述されるようにCH2状態からの入口で0と等しく、RH1状態に入った後に1レベルにセットされる。MSELRA及びMWE*信号は、ローアドレスが選択され、書込操作が指示されないようにハイレベルにセットされる。制御は、もしMRAS信号がメモリステートマシン600から真であり、RADHLD2信号即ちローアドレスホールド時間がセットされてより遅いメモリデバイスを指示するならばRH1状態からRH2に進行する。制御は、もしMRAS信号が真であり、RADHLD2信号がセットされないならば、RH1状態から、制御がRH2状態の次のクロック信号で進行する所でもあるCH1状態に直接進行する。他の全ての場として操作はRH1状態でループする。RH2状態において、RADRRDY信号及びCADRRDY信号はローにセットされ、MSELRA及びMWE*信号はハイ状態にセットされる。制御はRH2からCH1状態に進行する。

【0061】CH1状態において、RADRRDY信号は、ローアドレス確立時間が一致しなかったことを指示するためにローである。CADRRDY信号は、状態に入った直後にハイ即ち真値にセットされる。入口で、CADRRDY信号は、もし短CASアドレス確立時間(!CADSETUP2)が適切な読出又は書込毎にセットされたならばハイにセットされ、そうでなければローにセットされる。MSELRA信号は、カラムアドレスが選択されることを指示するために0にセットされ、MWE*信号は、もし書込操作が発生しているならばローである。

【0062】制御は、もしメモリステートマシン600がRN状態にあり、サイクルが中止され又はプロセッサ読出ページミスが待機中であるならばCH1状態からRAI状態に進行する。制御は、もしメモリステートマシンがRN状態になく、或いはサイクルが完全に中止されず、或いはプロセッサ読出サイクルがページヒットであるならばCH1状態からCH2状態に進行する。CH2状態において、RADRRDY及びCADRRDY信号はどんなアドレスも準備完了ができていないことを指示するためにローにセットされる。MSELRA信号はローにセットされ、MWE*はローにセットされて、書込操作が発生しカラムアドレスが形成されていることを指示する。制御は、メモリステートマシンがC3状態にあ

る時の実行中のバーストサイクル毎に或いはプロセッサ
読出ページヒット操作の待機中毎にCH2状態からCH
1状態に進行する。制御は、もしサイクルが中止され或
はアイドル状態が存在するならば、もしRAS*信号が
ハイにセットされるべきであるならばCH2状態からR
AI状態に進行する。制御は、もしプロセッサ読出ペー
ジミス操作が発生するべきであって、メモリステートマ
シン600がサイクルの終わりにあるならばCH2から
RH1状態に進行する。

【0063】それ故、メモリアドレスステートマシン6
00は、DRAMデバイスのためにMWE*信号を適切
に供給して、アドレスマルチプレクサ610を制御す
る。追加的に、RADRRDY及びCADRRDY即ち
列及びカラムアドレス準備完了信号は、メモリアイ
ングレジスタに供給されるように、カラム及びローア
ドレス確立時間に基づいてメモリステートマシン600に
供給される。

【0064】MD即ちメモリアドレスステートマシン6
14は図12で説明される。ステートマシン614は、リ
セット時にIDLE状態で始まる。制御は、中止されず
リフレッシュが発生していないプロセッサ又はPCIバ
ス98読出操作時にIDLE状態からRD状態に進行す
る。制御は、もしプロセッサ対メモリ書込操作が発生し
どんなリフレッシュもないならば或いはもしPCIバ
ス対メモリ書込操作が発生しどんなリフレッシュも再び
ないならば、DCD状態に進行する。さもなければ制御
はIDLE状態に留まる。制御は、もしメモリステートマ
シンがC3又はRN状態にあり、サイクルが中止され或
いはバーストサイクルでなく、それが33MHzプロセ
ッサによる書込操作であるならば、RD状態からDCD
状態に進行する。制御は、もしMRASがネゲートさ
れ、サイクルが中止され又はメモリステートマシンがC
3又はRN状態にあり、サイクルがバーストでなく又は
中止中で、非33MHzプロセッサ書込操作又はPCI
バス98書込サイクルが待機中で、どんなサイクルもア
クティブでないならば、RD状態からIDLE状態に進
行する。他の全ての場所で制御はRD状態に留まる。

【0065】制御は、もしMDSETUP2信号が真で
なくてこれが高速メモリアドレス確立場合であることを指
示し、メモリステートマシン600がC2状態になく、
33MHz操作がプロセッサのために示されるならば、
DCD状態からDH1状態に進行する。他の全ての場所
で制御はDCD状態からDS2状態に進行する。制御
は、もしプロセッサ200が33MHzで稼働中でなく
MDSETUP2信号がより遅いメモリアドレス確立時間
を指示するためにセットされたならば、DS2状態から
DS3状態に進行する。他の全ての場所で制御はDS2
からDH1状態に進行する。制御はクロック信号の次の
上昇縁でDS3状態からDS1状態に進行する。

【0066】DH1状態において、MDVALID信号

は、1にセット即ちアサートされて、カラムアドレス
ストロブへのメモリアドレス確立時間が一致したことを指
示する。これは、メモリ制御ステートマシン600をC
1状態に進行することを許容する。制御は、もし更なる
書込操作が待機中であるならば、DH1状態からDCD
状態に進行し、そうでなければDH2状態に進行する。
DH2状態において、MDVALID信号は、メモリが
もはや有効でないことを指示するために、0にセットさ
れる。MDVALID信号の値は、リセット時のクリア
を除いて、DH1及びDH2状態以外のどんな状態にお
いても変化しない。制御は、もしどんな書込操作もPC
Iバス98から或いはプロセッサ200から待機中でな
いならば、DH2状態からIDLE状態に進行する。他
の全ての場所で制御はDH2状態からDCD状態に進行
する。この様に、メモリアドレスステートマシン614
は、メモリアドレスが適切に設定されている時に、MDV
ALID信号をメモリステートマシン600に供給す
る。

【0067】プリチャージステートマシン602は図1
3に示される。リセット時にステートマシン602の操
作は、A状態から始まる。制御はMRAS信号がアサ
ートされない時にA状態に留まる。MRAS信号がアサ
ートされた時には、制御がプリチャージ時間が定義され
たクロック数に依存してB、C、D又はE状態の1つに
進行する。制御は、5クロック用のB状態、4クロック
用のC状態、3クロック用のD状態或いは2クロック
用のE状態に進行する。MRAS信号がアサートされな
い時には、制御がB状態からC、D、Eと序列的に進
行する。さもなければ制御は各特定の状態に留まる。その
後制御はMRAS信号がデアサートされる時に、E状態
からA状態に戻って進行する。それ故、プリチャージ
ステートマシン602は、特定のメモリ操作の開始時にA
状態を出発し、その後MRAS信号がネゲートされてプリ
チャージ期間が始まるまで、残りの状態を実行し始め
ない。これは、たとえバンクが連続メモリ操作でスイ
ッチされたとしても、もしリフレッシュ操作が無視され
るならば、所定のどのバンク毎の適当なプリチャージ
時間に帰着することが注目される。これは、上記注目さ
れるように、特定のバンクが選択されない時に、その
バンク用のRAS*信号がハイにセットされてプリチャ
ージ期間にあったので発生する。従ってもリフレッシュ
が選択されないならば、全てのケースで必要とされた
再充電を供給するのに十分な少なくとも1つのフルメモ
リサイクルを通してプリチャージされた。従って、プリ
チャージ時間は、現行のサイクルが動作している特定
のメモリバンクのそれ毎にセットされて、もし続けざ
まに起きるサイクルがその特定のメモリバンクで発生
するならば、メモリアドレスデバイスが適切にプリチャ
ージされる。

【0068】しかしながら、リフレッシュ操作は、好ま
しい実施例で知られていないように、リフレッシュサイ

クル後にどれかのバンクが要求されるであろうこの操作を幾分格下げして、それで好ましい実施例の設計を簡単にするために、すべてのDRAM形式毎のプリチャージ値が初期化又はPOSTソフトウェアによって最悪の場合を想定してセットされる。これはある場合性能を減少させるが設計を簡単にする。僅かにより複雑な設計は、好ましくはリフレッシュサイクル後の第1プリチャージ操作毎にのみ予め決定されたレジスタに供給された最悪値を用い、その後操作が各バンク毎に最適のタイミングに回復される。

【0069】書込アドレス及びデータは、配達されたデータを持つことに基づいてI2M行列410及び414を通して横切り（スタックされ）、アービタ550がメモリシステムによって行列410及び414へのアクセスを形成する。書込アドレスは、I2M行列410に置かれる時にPCONブロック400に供給され、PCONブロック400がL1及びL2キャッシュ制御器によるアドレスのスヌープ制御を許容する。読出操作は、書込操作と異なるファクションで発生する。読出アドレスは、読出要求が発行された指示を伴ってMCONブロック404に供給される。読出アドレスは、486マイクロプロセッサ用の4列32ビット語及びペンティアムプロセッサ用の8列32ビット語であるL2キャッシュラインの戻りアドレスに基づいて形成される。それ故、M2I読出が要求される時には4列或いは8列の32ビット語が形成され、ICONブロック402が読出操作からのデータを必要時にPCIバス98に適切に供給する。読出要求指示は、後述するメモリ対PCI次のアドレスステートマシンによって形成される。また、読出要求は、仲介用のアービタ550及びPCONブロック400に供給されて、L1及びL2キャッシュが修正ラインへのヒットの場合にライトバックとスヌープするのを許容する。アービタ550がPCIバス98読出と進行する許可を形成する時には、その後MCONブロック404が情報を処理することを経る。これは、I2M行列410及び414がどの待機中の書込操作も取出された後のみ発生して、データバッファ212に供給されるI2Mデータ行列414が指示された長さのブロック読出操作を行なうために用いることができることは、注目される。それ故、I2M行列410及び414が空にされる時に、読出操作は始まって、図14に示されるように、M2Iステートマシンによって制御される。ステートマシンは、リセット時にA状態が始まる。もしI2M行列414が空で、読出が要求され、L2キャッシュ208がフラッシュしていないならば、制御はB状態に進行する。これは通常の読出状態である。これの代りに、もしL2キャッシュ208がフラッシュされるならば、制御は、I2M行列414が空で、読出要求があると仮定するとA状態からF状態に進行する。さもなければ制御は、初期化されるべき読出操作を待つA状態に

留まる。

【0070】B状態から制御は4つの異なる状態の1つに進行する。制御は、もしM2IABORT信号が受信されてPCIバスマスタが読出操作を中止したことを指示したならば、或はIREQACK信号が真でアービタ550によって決定される次の操作がPCI読出であることを指示しSNPDONE信号が真でPCONブロック400がレベル2キャッシュ及びプロセッサ200のレベル1キャッシュへの読出操作アドレスをスヌープすることを完了したことを指示したならば、A状態に戻る。制御は、もしM2IABORT信号がアサートされず、次のサイクルがPCI読出であってアサートされたIREQACK信号によって指示され、PCONブロック400によって指示されるように修正された(SNP HITM)位置へのスヌープヒットでなく、スヌープ操作がPCONブロック400によっても指示されるように完了しなかったならば、B状態からC状態に進行する。これは、読出操作が要求されて仲介されたが、スヌープ操作が行なわれず修正された位置へのヒットがまだ発生するかもしれないので、中止されるべきでない場合である。制御は、もし操作が中止されず、PCI読出が仲介されなかったが、修正された位置へのスヌープヒットがあり、ライトバック操作がまだ完了しなかったならば、B状態からD状態に進行する。制御は、もしサイクルが中止されずIREQACK信号によって指示されるように仲介され、適切なキャッシュ制御器からのライトバック操作が発生しているように修正された位置へのスヌープヒットがあるならば、B状態からE状態に進行する。他の全ての場合においては、制御がB状態に留まる。

【0071】制御は、もしサイクルが中止されるならば或いは、もしスヌープが修正された位置へのヒットなしで完了されるならばC状態からA状態に戻って進行する。制御は、もしサイクルが中止されずに修正された位置へのスヌープヒットがあったならば、C状態からE状態に進行する。さもなければ、制御は、スヌープ操作が完了され、サイクルが中止され、或は修正ラインへのスヌープヒットがあるまでC状態に留まる。制御は、もし操作が中止されるならば、或いは指示時にそのPCIサイクルが次の操作でありスヌープが完了したならばD状態からA状態に戻って進行する。これは、ライトバックがPCI操作に取って代るように、ライトバックが完了され、その後PCI操作が次に発生した後に発生する。制御は、もしそれが中止されず、PCI要求が次にあり、スヌープが十分に完了されないならば、D状態からE状態に進行する。さもなければ制御はD状態に留まる。

【0072】E状態において、CLR I2M信号は、メモリ制御器210の他の部分に、ライトバックの故にI2M行列414中のデータがフラッシュされて廃棄しな

なければならないことを指示するためにセットされる。CLR I 2MDONE 信号は、このフラッシュが完了したことを指示する。制御は、もしサイクルが中止されず、信号CLR I 2MDONE 即ち行列実行クリア信号が真であり、スヌープがまだ完了しないならば、E 状態から F 状態に進行する。制御は、もしサイクルが中止されず、行列 414 のクリアが完了され、スヌープが実行されたならば、E 状態から G 状態に進行する。制御は、もしサイクルが中止されるならば、E 状態から A 状態に進行し、他の全ての場合 E 状態に留まる。

【0073】制御は、もしそれが中止されずスヌープサイクルが完了されたならば、F 状態から G 状態に進行する。他の制御は、もしサイクルが中止されるならば、F 状態から A 状態に戻って進行する。さもなければ制御は F 状態に留まる。制御は、もしサイクルが中止されるならば或いはもし IREQACK 信号によって示されるようにそれが次のラインならば、G 状態から A 状態に戻って進行する。

【0074】それ故、M2I ステートマシンは、バッファ 212 に配置された I 2M 行列 414 を用いてメモリ対 PCI インタフェイスからの情報の転送を制御する。データは、行列 414 がどの書込データもクリアされ、その後スヌープ情報及びアビタ 550 が進行できる時にのみ基づいて進行するまで転送されない。

【0075】上記注目されるように、好ましい実施例の設計は、PCI メモリ読出多重操作が受信された時に先読み操作を行なう。操作は、次に述べる通り進行する。

ICON ブロック 402 は、PCI バス 98 からメモリ読出多重サイクルを受信し、I 2M 行列 414 に空きがある時にアドレスと共に M2I 読出サイクル要求を MCON ブロック 404 に出力する。MCON ブロック 404 は上述のように仲介して、最終的に M2I 読出要求を始める。このブロックは、操作を始めた時に、次のアドレス即ち M2INA 信号を ICON ブロック 402 に供給して、適切な準備完了信号と共にデータを I 2M 行列 414 に供給する。ICON ブロック 402 は、メモリ読出多重命令が発生していることを知り、M2INA 信号が受信され、I 2M 行列 414 がデータの他のキャッシュラインを受信することができる時に次のアドレスで他の M2I 読出サイクル要求を出す。ICON ブロック 402 も MCON ブロック 404 から準備完了信号を受信して、PCI バス 98 にデータを供給する。MCON ブロック 404 は M2I 読出サイクル要求を受信して、サイクルが仲介を得た時にそれを実行する。これは、ICON ブロック 402 が初期読出要求毎に I 2M 行列 414 から全データを取去る前に月並みにある。その後 MCON ブロック 404 は読出要求を始めて、他の M2INA 信号を出す。その後 MCON ブロック 404 は、I 2M 行列 414 にデータを供給する。ICON ブロック 402 は、この M2INA 信号を受信して、他のキャッ

シュラインを受信するために I 2M 行列 414 に利用できるスペースがあるかどうかをわかるために再び調べる。ICON ブロック 402 が PCI バス 98 に十分なデータを供給したので空きがある時には、次の M2I 読出要求が MCON ブロック 404 に供給される。このプロセスは、メモリ読出多重が完了し、ページの境界が越え、或いは PCI バスマスタがサイクルを中止するまでも続行される。

【0076】中止場合は、待機中の先読み操作をできるだけ早く終わらせて全体のキャッシュラインの回復をセーブするので関心の 1 つである。これは、MEMSM600、MADRS608、MDCDSM614 及び M2I ステートマシンの説明で理解することができる。この迅速な終端は、C3 及び RN 状態から RI 又は C1 状態への戻りのように MEMSM600 で理解されて、現行の個々の読出操作が完了した途端サイクルが終わり、従ってフルキャッシュライン読出の完了前に潜在的に終了する。同様に、MADRS608 は、もしサイクルが始まらなかったならば或いはカラムアドレスが供給された時に RA1 状態に戻る。MDCDSM614 は、もしどんな操作も始められなかったならば或いはもし MEMSM600 が C3 又は RN 状態にあるならば IDLE 状態に戻る。M2I ステートマシンは、M2I ABORT 信号が受信される時に A 状態に戻る。中止の検知において、ICON ブロック 402 は、読出サイクルの終わりを決定し、どんなデータもないことを指示するために I 2M 行列 414 中のデータへのそのポインタをリセットし、従って先読データを効果的に捨てている。従って先読み操作は、中止指示を受信した直後に終了して、フルキャッシュライン回収を完了しないことによって時間を節約する。

【0077】先読できる次のアドレス指示は、図 15 に示された M2INA 即ちメモリ対 PCI 次のアドレスステートマシンによって形成される。初期読出アドレスは従来の読出要求メカニズムを使って形成された。M2INA 信号は、次の読出サイクルが存在できることを指示するために、MCON ブロック 404 に供給される。このステートマシンは、リセット時に A 状態から操作を始め、もし PCI 読出操作が IREQACK 信号によって示されるように次であり中止されないならば、B 状態に進行する。さもなければ制御は A 状態に留まる。制御は、もしサイクルが中止されているならば、或いはもし I 2M 行列 414 がライトバックの為にクリアされるべきであるならば、或いはもしスヌープが完了され M2I ACK 信号が供給されて、先の読出操作が MCON ブロック 404 で確認されること指示するならば、B 状態から A 状態に戻って進行する。さもなければ制御は B 状態に留まる。制御は、もしサイクルが中止されず、行列をクリアする必要がなく、スヌープ操作が完了しなくて、それなのに M2I ACK 信号が受信されたならば、B 状

態からC状態に進行する。制御は、もしサイクルが中止され、I2M行列414がクリアされ、或いはスヌープが完了したならば、C状態からA状態に進行し、さもなければC状態に留まる。従ってM2INAステートマシンは、中止指示の受信時にアイドルに戻る。

【0078】M2INA信号はMCONブロック404に供給されて、次のアドレスが供給され、即ち、他の読出要求がI2M行列414にPCIバス98の前に満たし続けるために発行することができることを指示する。M2INA信号は、もしサイクルが中止されずI2M行列414がクリアされず前の読出サイクルのスヌープが完了しM2INAステートマシンがC状態或いはB状態にありM2IACK信号が受信されたならば、出力される。このM2INA信号は、先のアドレスの処理がMCONブロック404によって完了し、次の読出アドレスの処理が始まることができる指示である。読出アドレス値の実際のインクリメントは、8ビットカウンタを用いたICONブロック402で行なわれ、従って全体的な先読み長を256のアドレス値に制限している。カウンタが255に到達した時に、先読み操作は、メモリ読出多重を未接続にさせることによって単純さのために説明されなかったロジックによって終端する。新規アドレスは、メモリ読出多重操作を続けるために、PCIバスマスタから受信されなければならない。

【0079】それ故、好ましい実施例によるメモリ制御器は、多くの改善を含む。それは、非常に柔軟で、複数のプロセッサ及びメモリデバイス速度との使用で単純でまだ高性能を許容し、柔軟性及び規模の経済性を許容する。それは、プロセッサ及びPCIバス状態でのアイドル状態のRAS*信号のレベル予測を基準にして改善された性能を提供する。それは、幾つかのバスから非常に深い書込ポイントを許して、容易に種々のコヒーレンシ問題を扱う。また、それは、基本的な読出操作が中止された時に、先読みサイクルを迅速に終了する速い方法を提供する。

【0080】発明の前述の開示及び記述がその実例で説明的である。サイズ、形、材料、構成要素、回路要素、配線接続及びコンタクトの種々の変形が、説明された回路の詳細及び構成及び操作方法と同様に発明の精神を逸脱しないで作られてもよい。

【図面の簡単な説明】

【図1】好ましい実施例によるコンピュータシステムの

システム基板のブロック図である。

【図2】図1のコンピュータシステムと使用される本発明によるプロセッサ基板のブロック図である。

【図3】図1のコンピュータシステムと使用されるビデオのシステムのブロック図である。

【図4】図2のプロセッサ基板のメモリ制御器のブロック図である。

【図5】図2のデータバッファのブロック図である。

【図6】図4のメモリ制御器における特定のメモリサイクルのためにメモリタイミング情報及びアドレスを形成するために使用された回路のブロック図である。

【図7】図4のメモリ制御器の部分のブロック図である。

【図8】図4のメモリ制御器の部分のブロック図である。

【図9】図4のメモリ制御器のCAMヒット検知ロジックのブロック図である。

【図10】図4のメモリ制御器の種々の部分の回路の操作毎のステートマシン図である。

【図11】図4のメモリ制御器の種々の部分の回路の操作毎のステートマシン図である。

【図12】図4のメモリ制御器の種々の部分の回路の操作毎のステートマシン図である。

【図13】図4のメモリ制御器の種々の部分の回路の操作毎のステートマシン図である。

【図14】図4のメモリ制御器の種々の部分の回路の操作毎のステートマシン図である。

【図15】図4のメモリ制御器の種々の部分の回路の操作毎のステートマシン図である。

【符号の説明】

200 プロセッサ

202 データバス

204 アドレスバス

206 制御バス

208 L2キャッシュメモリ

210 DRAM制御器(CMC)

214 バンク構成のメモリ

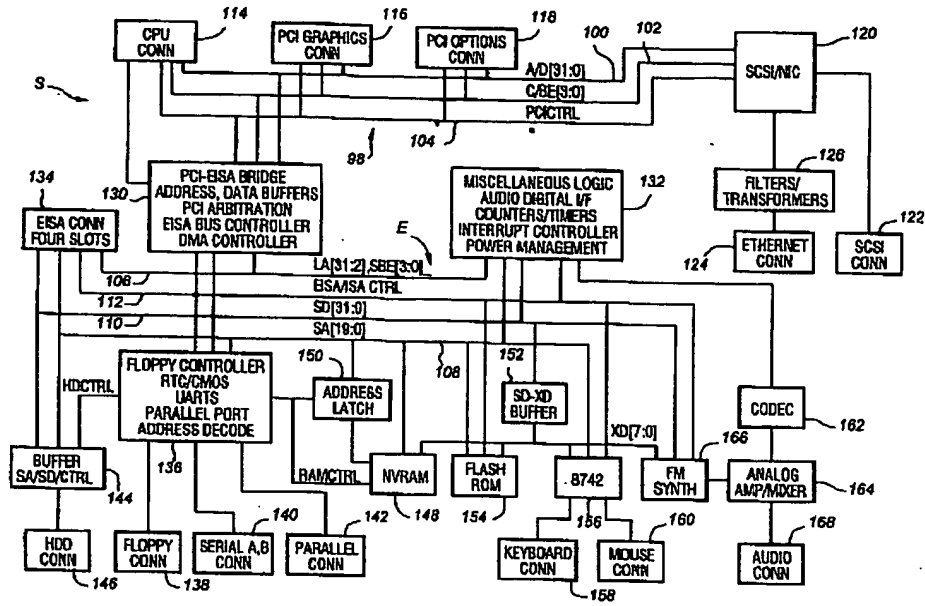
216 メモリアドレスバス

218 メモリデータバス

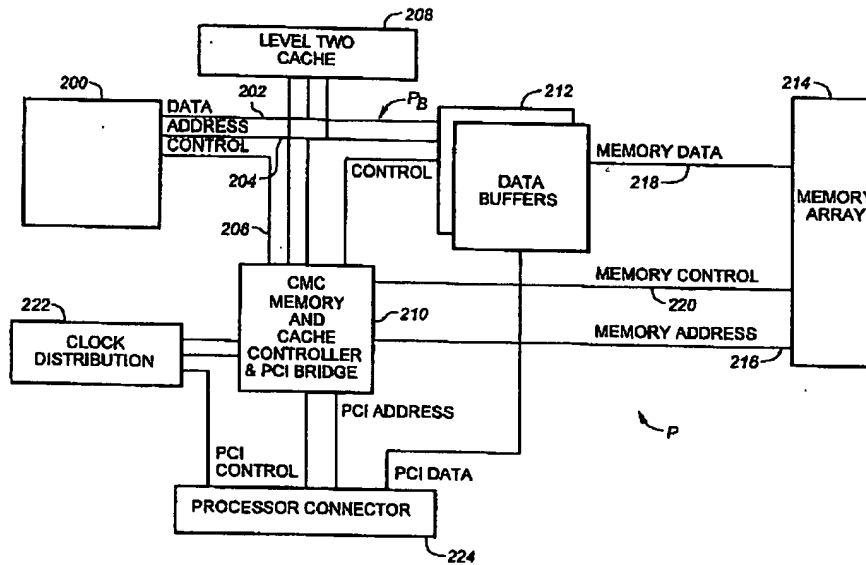
220 メモリ制御バス

222 クロック分配回路

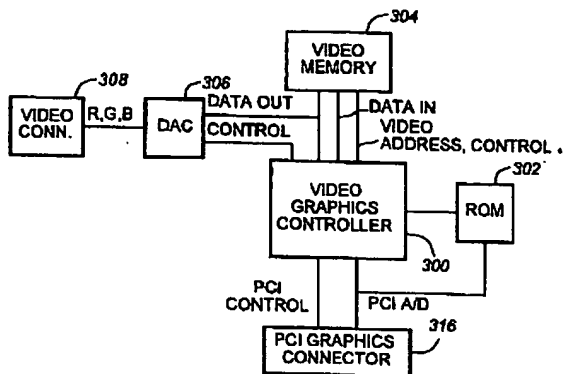
【図 1】



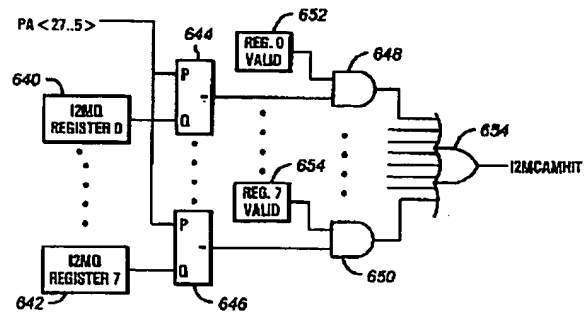
【図 2】



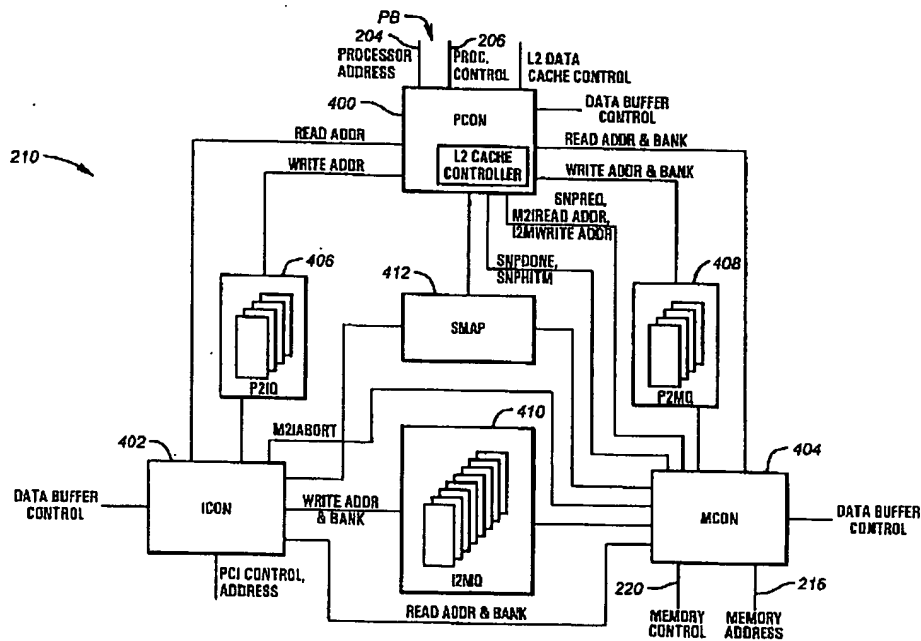
【図 3】



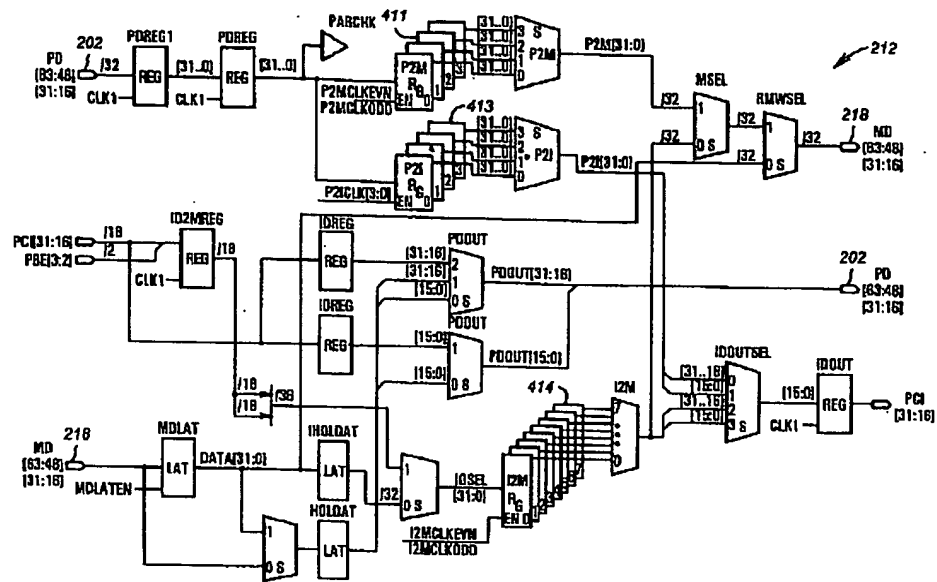
【图9】



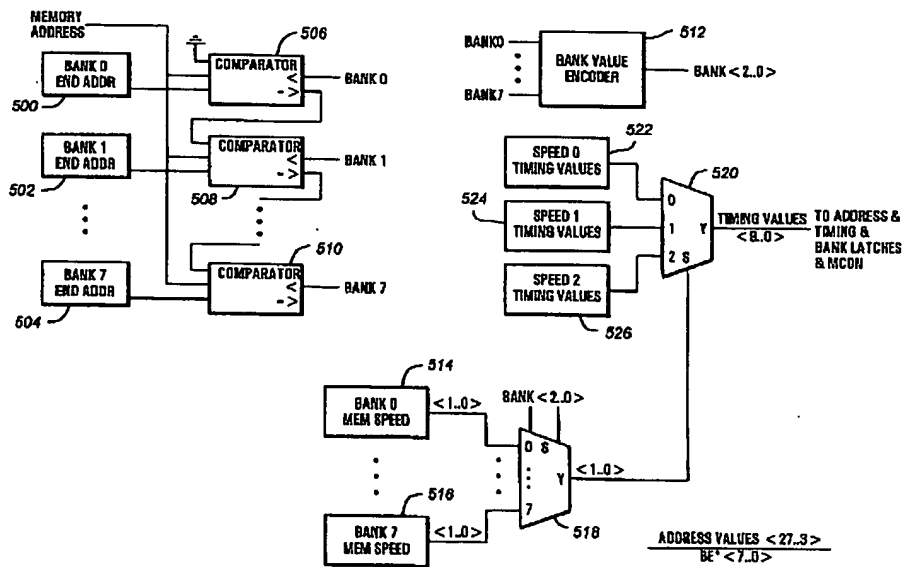
【図 4】



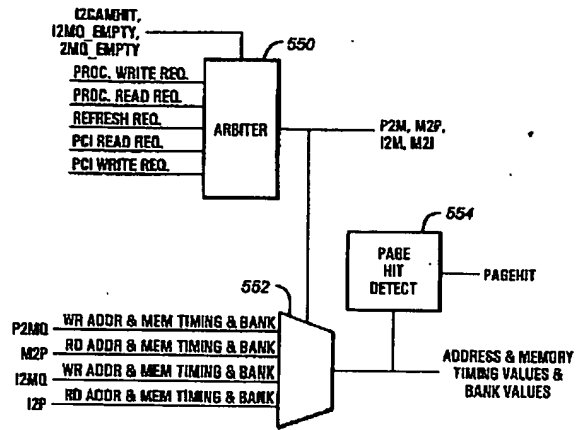
【図 5】



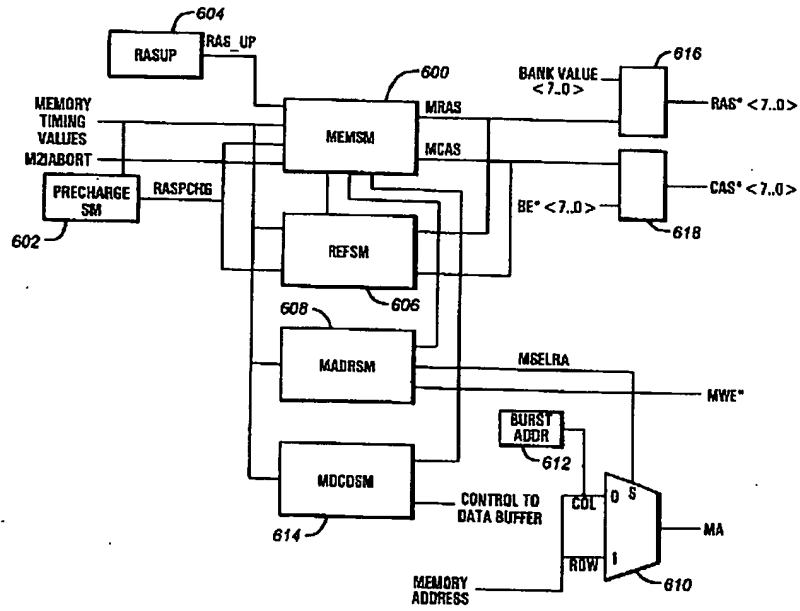
【図 6】



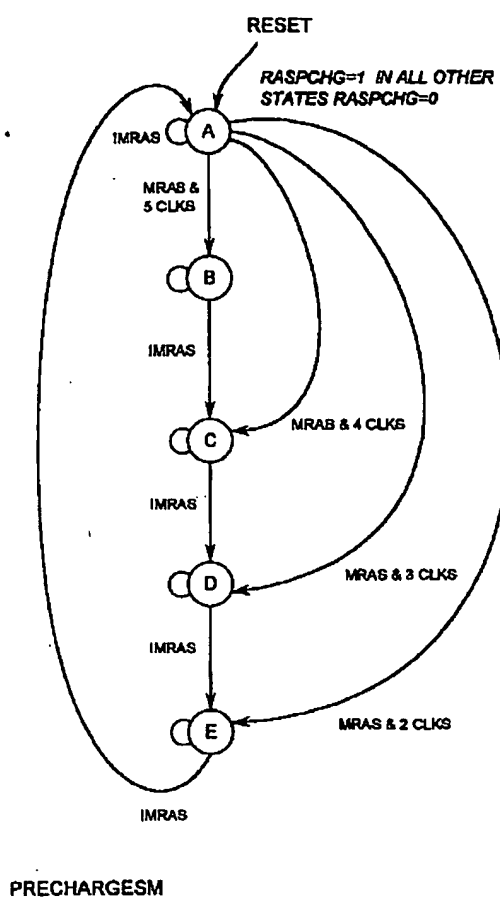
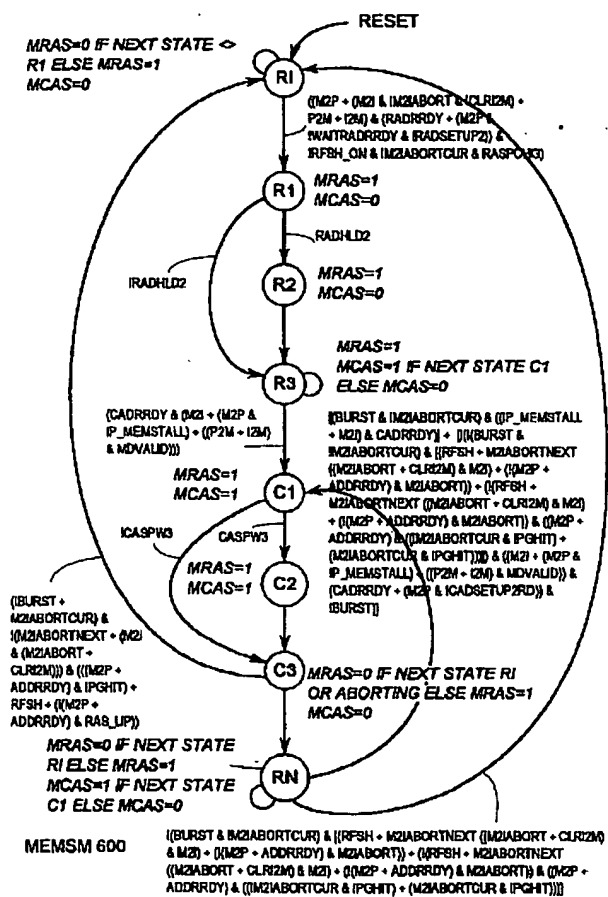
【図 7】



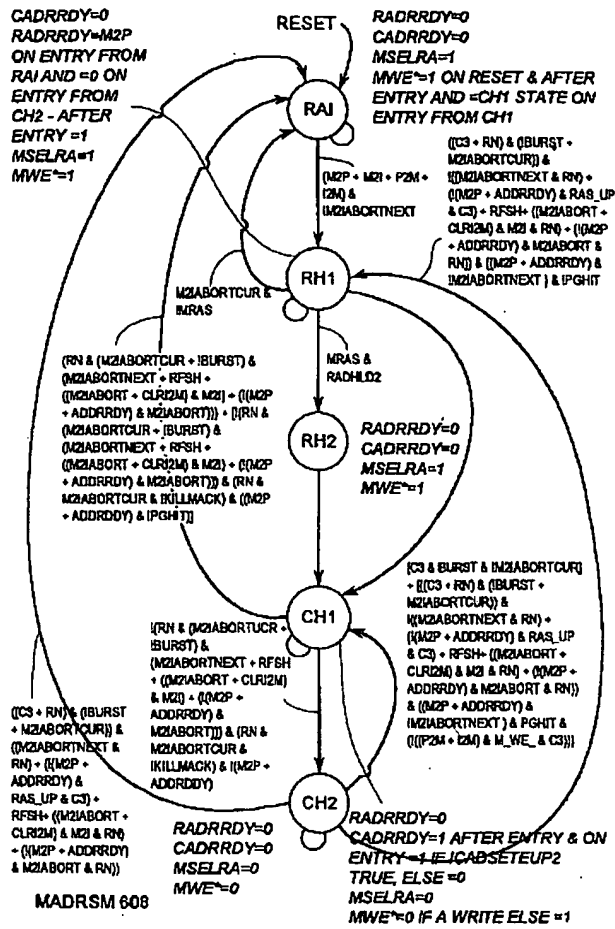
【図 8】



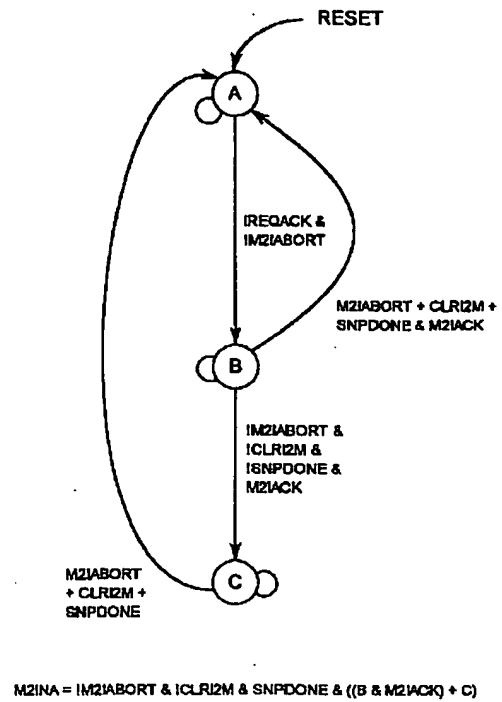
【图 13】



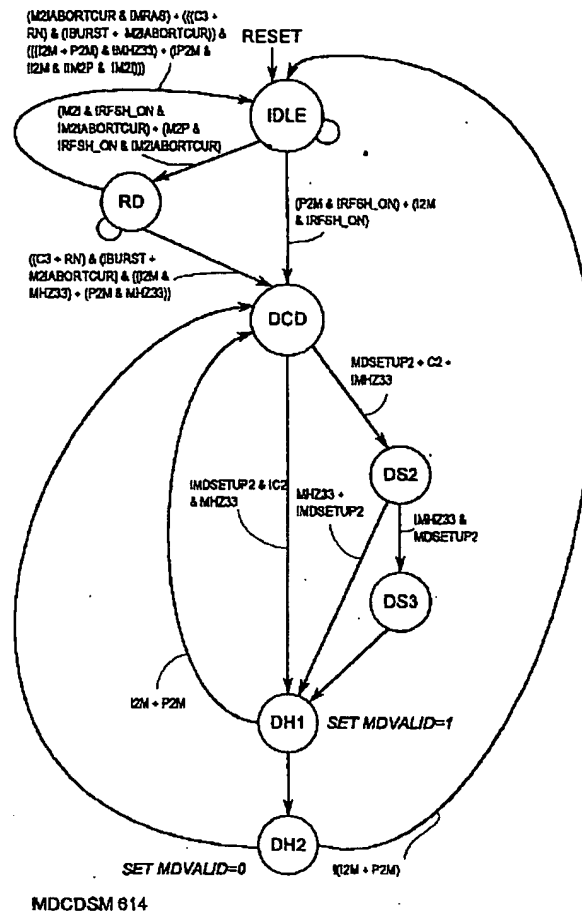
【図 11】



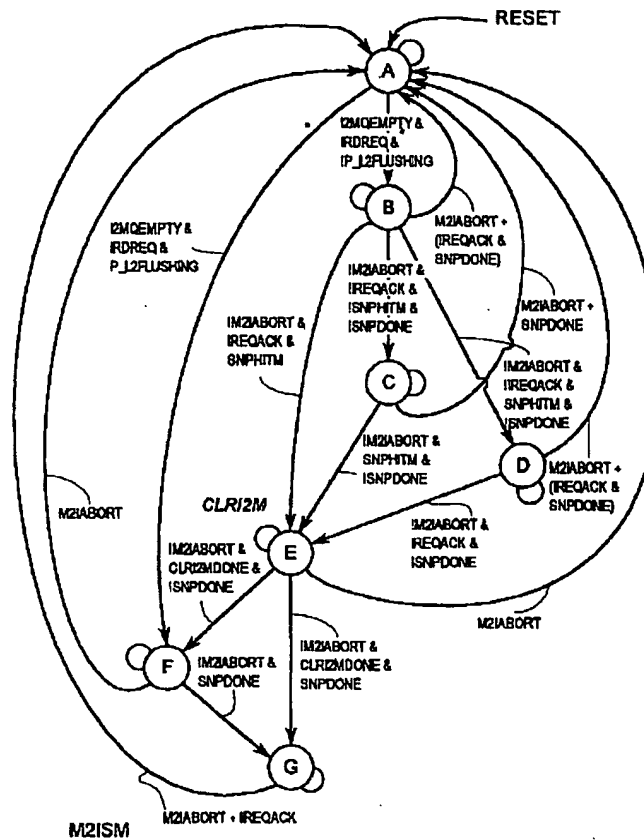
【図 15】



【図 12】



【図 14】



フロントページの続き

(72)発明者 ジョン・イー・ラーソン
 アメリカ合衆国77449テキサス州ケイ
 ティ・フrint・ヒル・ドライブ18354
 (72)発明者 ゲアリ・ダブリュー・トム
 アメリカ合衆国77375テキサス州トムボ
 ール・マラット・コート16406

(72)発明者 マイケル・ジェイ・コリンズ
 アメリカ合衆国77375テキサス州トムボ
 ール・レイクストーン・ドライブ16030
 (72)発明者 マイケル・モライアティ
 アメリカ合衆国77379テキサス州スプリ
 グ・サニイ・ポイント・ドライブ8830

【外国語明細書】

1. Title of Invention

EASILY PROGRAMMABLE MEMORY CONTROLLER WHICH CAN ACCESS DIFFERENT
SPEED MEMORY DEVICES ON DIFFERENT CYCLES

2. Claims

1. A dynamic random access memory controller for use with a plurality of dynamic random access memories controller receiving addresses and cycle timing information from a bus, the dynamic random access memory controller comprising:

means for determining which bank of dynamic random access memories is to be accessed for a received address;

means for indicating a dynamic random access memory type for each bank;

means for storing a plurality of dynamic random access memory timing parameters for portions of dynamic random access memory cycles for each dynamic random access memory type;

means for determining the plurality of dynamic random access memory timing parameters applicable to said received address based on the bank determination and the dynamic random access memory type indication for the determined bank; and

means receiving the determined plurality of dynamic random access memory timing parameters, the determined bank and the address for providing row and column addresses and address strobes to the dynamic random access memories according to the determined plurality of dynamic random access memory timing parameters for that particular bank.

2. The dynamic random access memory controller of claim 1, wherein

the dynamic random access memory controller further receives a clock signal and wherein said means for providing row and column addresses and address strobes operates synchronously based on the received clock signal.

3. The dynamic random access memory controller of claim 1, wherein said means for providing row and column addresses and address strobes provides the row address strobe to the banks of dynamic random access memories other than the determined bank at a negated level.

4. The dynamic random access memory controller of claim 1, wherein the dynamic random access memories are page mode devices and wherein said means for providing row and column addresses and address strobes includes a page bit detector and provides the row and column addresses and address strobes in page mode if a page bit is detected.

5. The dynamic random access memory controller of claim 1, further comprising:

means for indicating the address range assigned to each bank of dynamic random access memories.

6. The dynamic random access memory controller of claim 1, wherein said plurality of dynamic random access memory timing parameters include the row address hold time from the row address strobe, the column address setup time to the column address strobe, the row address setup time to the row address strobe and the column address strobe width.

7. The dynamic random access memory controller of claim 6, wherein

said column address time setup parameter includes separate parameters for read and write operations.

8. A computer system, comprising:

a plurality of dynamic random access memories arranged in a plurality of banks;

a processor providing memory addresses and cycle timing information;

a bus from transferring said processor memory address and cycle timing information;

a dynamic random access memory controller receiving address and cycle timing information from said bus, said dynamic random access memory controller including:

means for determining which bank of dynamic random access memories is to be accessed for a received address;

means for indicating a dynamic random access memory type for each bank;

means for storing a plurality of dynamic random access memory timing parameters for portions of dynamic random access memory cycles for each dynamic random access memory type;

means for determining the plurality of dynamic random access memory timing parameters applicable to said received address based on the bank determination and the dynamic random access memory type indication for the determined bank; and

means receiving the determined plurality of dynamic random access memory timing parameters, the determined bank and the address for providing row and column addresses and address strobes to the dynamic random access memories according to the determined plurality of dynamic random access memory timing parameters for that particular bank.

9. The computer system of claim 8, further comprising:
means for providing a clock signal, and
wherein said dynamic random access memory controller further receives said clock signal and wherein said dynamic random access memory controller means for providing row and column addresses and address strobes operates synchronously based on the received clock signal.

10. The computer system of claim 8, wherein said dynamic random access memory controller means for providing row and column addresses and address strobes provides the row address strobe to the banks of dynamic random access memories other than the determined bank at a negated level.

11. The computer system of claim 8, wherein the dynamic random access memories are page mode devices and wherein said dynamic random access memory controller means for providing row and column addresses and address strobes includes a page hit detector and provides the row and column addresses and address strobes in page mode if a page hit is detected.

12. The computer system of claim 8, wherein said dynamic random access memory controller further includes:

means for indicating the address range assigned to each bank of dynamic random access memories.

13. The computer system of claim 8, wherein said plurality of dynamic random access memory timing parameters include the row address hold time from the row address strobe, the column address setup time to

the column address strobe, the row address setup time to the row address strobe and the column address strobe width.

14. The computer system of claim 13, wherein said column address time setup parameter includes separate parameters for read and write operations.

3. Detailed Explanation of the Invention

The invention relates to memory controllers utilized in computer systems, and more particularly to a memory controller which can operate with different speed memory devices in separate banks and additionally with different speed microprocessors.

Computer systems are becoming ever more powerful by the day. Users are requiring more capabilities for ever more complicated and sophisticated applications and computer system manufacturers are responding. Computer speeds have dramatically increased over the last number of years so that now desktop and file server computers can readily outperform mainframe computers of 10-15 years ago. But the quest for further performance is never ending. To this end, the microprocessor manufacturers have been developing ever faster microprocessors.

However, a computer system is far more than just a microprocessor. There are many other subsystems that must cooperate with the microprocessor to provide a complete computer system. It is desirable to optimize as many of these subsystems as possible and yet take into account cost and system flexibility to satisfy varied user desires.

Two of the subsystems which have not maintained pace with the development of microprocessor are the main memory systems and the

input/output buses. Main memory system shortcomings have been much alleviated by the use of cache memory systems, but in the end all memory operations must ultimately come from the main memory, so that its performance is still a key piece in the overall performance of the computer system. Many advanced memory architectures and techniques have developed over the years. One of the most common techniques is the use of paged mode memory devices or DRAMs, where the actual memory address location value is divided into rows and columns, and if the row address, i.e., the page, is the same for the subsequent operation, only column addresses need to be provided to the DRAM. Although there is a certain amount of overhead required, it easily pays for itself by the improved performance gained during a page hit. So basic page mode operation provides a major performance increase, but more performance is always desired.

One further performance increase relates to an improvement for determining the level of the row address strobe or RAS* signal when the memory system is idle. As is well known, the RAS* signal must be negated or set high to allow a new page or row address to be provided and there is also a precharge time requirement. Thus, there is a performance penalty if the RAS* signal is raised when the next operation is actually a page hit. Similarly, there is a delay if the RAS* signal is kept low and the operation is a page miss, as the full precharge time must also be expended after the cycle has been issued. To address this concern, various techniques have been developed to predict whether the RAS* signal should be kept low or should be returned high to indicate a new page cycle. The prediction can be done several ways, as indicated in Serial No. 08/034,104 filed March 22, 1993, entitled "Memory Controller That Dynamically Predicts Page Misses." In that application several techniques are used. A first, simple technique bases the prediction on

the type of the last cycle performed by the processor, with the choice always fixed. A second, more sophisticated technique samples the hits and misses for each cycle type and then sets the RAS* level based on this adaptive measurement. But the techniques have been based on the use of the processor cycles and have not based themselves on the I/O bus cycles. Therefore, I/O bus master operations still performed at lesser levels.

One high performance I/O bus is the PCI or Peripheral Component Interconnect bus developed by Intel Corp. and accepted by many computer manufacturers. PCI is a high performance bus and allows numerous bus masters to be present. The bus masters are essentially local processors which perform specific duties, not general processing duties. By having these bus masters, the main processor is able to off load various specialized processing tasks, so that more tasks can be performed in parallel, thereby increasing the performance of the computer system. This is but one example of how parallelism is being used in current computer systems.

It is desirable to have as many operations running in parallel or concurrently as possible to allow increased overall performance. One way this concurrent operation has been done in the past is by the use of write posting, where a single cycle from the processor is latched into a posting buffer and ready is returned to the processor prior to the write cycle actually being completed to the memory or I/O device. The entire data and address values are posted in a latch and then the cycle executed on the target bus when possible. However, write posting has been kept at a very simple level, such as one level per bus, because of complications in memory coherency and cycle ordering which result if deeper posting were to be performed. Therefore, it can be seen that there are numerous gains that could be obtained if one were able to write post more than a single operation to a given bus, if the complications

could be simply solved.

The PCI bus provides opportunities to increase overall system performance, particularly that of the memory system. One of the read operations defined for the PCI bus is what is termed as a Memory Read Multiple cycle, which is used to indicate a desire to read a number of cache or memory lines, not just a single line. As noted in the PCI bus cycle definition, this cycle decoding provides an opportunity for the memory controller to start doing read aheads or pipelining so that the data can be obtained prior to actually being required on the PCI bus. With this data then obtained, the memory controller can allow access by the processor, thus further increasing overall system concurrency. However, it is also common for PCI bus masters to abort cycles prior to their completion, and if such an aborted cycle were to occur shortly after a Memory Read Multiple cycle has commenced, then a read ahead operation would have been started and would conventionally complete, only to have the data then immediately discarded. This would reduce overall system performance because of the wasted operations needed to start and complete the full read ahead operation. Therefore, it would be desirable to reduce the wasted time when doing read aheads during Memory Read Multiple cycles when the cycle is aborted early by the bus master on the PCI bus.

Further, personal computer systems are becoming mass market products, and therefore need to be very flexible to meet the widely varying particular goals of users. For example, some users may desire the ultimate in performance with little regard for cost, whereas other users may be significantly more cost sensitive. One area where cost directly impacts performance is in the speed of the memory devices used in the main memory. Another area of impact is the economies of scale which could be obtained by using a single memory controller chip for many

different microprocessor configurations and speeds. But using a single memory controller usually involves performance tradeoffs. The memory controller as disclosed in U.S. Patent No. 5,333,293 addressed the multiple speed processor point, but could use only a single speed of memory devices, thus limiting user options and performance tradeoffs. In another memory controller as disclosed in Serial No. 08/034,290 filed March 22, 1993, the memory controller can handle different speed memory devices on a bank by-bank or module-by-module basis, and yet allows optimal timing for each particular memory device. However, this memory controller was designed to be used with a single processor operating at a single speed, thus providing user flexibility but not economy of scale. It would be more desirable to allow numerous types and speeds of processors to be utilized with a single memory controller, and yet allow use of numerous types and speeds of memory devices without requiring great complexity.

A computer system according to the present invention has a memory controller that provides numerous performance increases, particularly in the PCI bus environment, and can readily work with numerous types and speeds of processors and different speed memory devices.

The memory controller provides a series of queues between the processor and the PCI bus and the memory system to allow deep write posting. In the preferred embodiment, four quadword addresses can be posted from the processor and eight quadword addresses from the PCI bus for write operations. Memory coherency is maintained in two different ways. Before any read operations are accepted from the PCI bus, both of the posting queues must be empty. In this way, all writes are completed prior to the read occurring, so that the main memory is coherent for the read operation from the PCI bus. However, more performance is desired

from the processor, and therefore to maintain coherency a content addressable memory (CAM) is utilized as the PCI to memory queue. When the processor performs a read request, the CAM is checked to determine if one of the pending write operations in the PCI to memory queue is to the same address as that read operation of the processor. If so, the read operation is not executed until the PCI memory queue has cleared that entry. If no address hit occurs, the read operation is accepted and executed according to arbitration priority rules. Again, in this manner, the main memory is coherent prior to the read operation occurring. It is noted that allowing two write operations to the same address to be present in the two queues is not a problem and does not produce incoherent results, as the exact timing between the buses would never be clear in any event.

In the preferred embodiment the PCI bus capability of read ahead operations when a Memory Read Multiple has been requested is present. This allows the memory system to obtain data at a high rate and leave it posted for reading by the PCI bus master when indicated by the particular cycle. However, as noted in the background, it is possible that the PCI bus master would abort the cycle prior to its completion. To resolve this problem, a memory controller according to the preferred embodiment receives an abort signal from the PCI bus interface and as soon thereafter as can be done, while maintaining DRAM data integrity, terminates the read ahead cycle, even though the read ahead cycle has not fully completed. Thus, the read ahead cycle is aborted as soon as possible. Therefore, the full read ahead does not occur, so that the situation of an abort occurring during a read ahead operation does not overly hinder performance as would normally be the case.

To further improve the system, the memory controller of the preferred embodiment has improved prediction rules for determining when

to precharge the DRAM devices. The prediction rules are based on whether the cycle is coming from the processor or is coming from the PCI bus. By using these new rules, more efficient precharging is done, and additionally, more page mode cycles can be performed than otherwise would have been done according to the prior art.

Finally, the memory controller of preferred embodiment is highly programmable for multiple speeds and types of processors and several speeds of memory devices, and yet can be simply programmed. The memory controller includes a plurality of registers that specify the number of clock periods for the particular portions of a conventional DRAM cycle, such as the address set up and hold times, CAS* signal pulse width, the precharge time and the data set up time. These registers are thus consistent with the normal timing concerns and parameters of DRAM devices, so that the designer need only know the particular processor type and clock speed and memory device speed and then the registers can be properly and simply programmed from a small matrix or table. Complex operations and alternatives are not necessary and the clock period values can be easily determined. By the use of the single memory controller for multiple processor types and speeds and plural speeds, the economies of scale can now be obtained by increasing manufacturing volumes while still allowing user flexibility.

The following disclosures are hereby incorporated by reference:

U.S. application Serial No. 08/324,016, entitled "Single Bank, Multiple Way Cache Memory," by Alan L. Goodrum, Jens K. Ramsey, and Michael J. Collins, filed October 14, 1994;

U.S. Application Serial No. 08/324,246, entitled "Memory Controller With Write Posting Queues for Processor and I/O Bus Operations and Ordering Logic for Controlling the Queues." by Michael J. Collins, Gary

W. Thome, Michael Moriarty, Jens K. Ramsey, and John E. Larson, filed October 14, 1994;

U.S. Application Serial No. 08/323,263, entitled "Data Error Detection and Correction System," by William J. Walker and Alan L. Goodrum, filed October 14, 1994;

U.S. Application Serial No. 08/324,020, entitled "Circuit for Invalidating Portions of a Cache Memory if a Write Occurs to a Write Protected Area of Main Memory" by Jens K. Ramsey, filed October 14, 1994

U.S. Application Serial No. 08/323,110, entitled "Circuit for Placing a Cache Memory Into Low Power Mode in Response to Special Bus Cycles," by Jens K. Ramsey, and Jeffrey C. Stevens, filed October 14, 1994; and

U.S. Application Serial No. 08/324,011, entitled "System for Arbitrating Access to Memory" by John E. Larson, Michael Moriarty, Michael J. Collins and Gary W. Thome, filed October 14, 1994; all of which are assigned to the assignee of this invention.

Referring now to Figure 1, the system board S of a computer system according to the present invention is shown. In the preferred embodiment, the system boards contain circuitry and slots for receiving interchangeable circuit boards. In the preferred embodiment, there are two primary buses located on the system board S. The first bus is the PCI or Peripheral Component Interconnect bus 98 which includes address/data portion 100, control and byte enable portion 102 and control signal portion 104. The second primary bus on the system board S is the EISA bus E. The EISA bus E includes LA address portion 106, SA address portion 108, SD data portion 110 and EISA/ISA control signal portion 112. The PCI and EISA buses P and E form the backbones of the system board S.

A CPU connector 114 is connected to the PCI bus 98 to receive interchangeable processor cards, such as the one shown in Figure 2. A

PCI graphics connector 116 is connected to the PCI bus 98 to receive a video graphics card as shown in Figure 3. A PCI option connector 118 is also connected to the PCI bus 98 to receive any additional cards designed according to the PCI standard. In addition, a SCSI and network interface (NIC) controller 120 is connected to the PCI bus 98. Preferably the controller 120 is a single integrated circuit and includes the capabilities necessary to act as a PCI bus master and slave and the circuitry to act as a SCSI controller and an Ethernet interface. A SCSI connector 122 is connected to the controller 120 to allow connection of various SCSI devices, such as hard disk drives and CD-ROM drives. An Ethernet connector 124 is provided on the system board S and is connected to filter and transformer circuitry 126, which in turn is connected to the controller 120. This forms a network or Ethernet connection for connecting the system boards and computer to a local area network (LAN).

A PCI-EISA bridge 130 is provided to convert signals between the PCI bus 98 and the EISA bus E. The PCI-EISA bridge 130 includes the necessary address and data buffers and latches, arbitration and bus master control logic for the PCI bus, EISA arbitration circuitry, an EISA bus controller as conventionally used in EISA systems and a DMA controller. Preferably the PCI-EISA bridge 130 is a single integrated circuit, but other combinations are possible. A miscellaneous system logic chip 132 is connected to the EISA bus E. The miscellaneous system logic chip 132 contains a digital audio interface, counters and timers as conventionally present in personal computer systems, an interrupt controller for both the PCI and EISA buses P and E and power management logic, as well as other miscellaneous circuitry. A series of four EISA slots 134 are connected to the EISA bus E to receive ISA and EISA adapter cards. A combination I/O chip 136 is connected to the EISA bus E. The

combination I/O chip 136 preferably includes a floppy disk controller, realtime clock (RTC)/CMOS memory, two UARTs, a parallel port and various address decode logic. A floppy diskconnector 138 for receiving a cable to a floppy diskdrive is connected to the combination I/O chip 136. A pair of serial port connectors are also connected to the combination I/O chip 136, as is a parallel port connector 142. A buffer 144 is connected to both the EISA bus E and the combination I/O chip 136 to act as a buffer between the EISA bus E and a hard disk drive connector 146 to allow connection of an IDE-type hard disk drive. A non-volatile random access memory (NVRAM) 148 is connected to the EISA bus E and receives its control signals from the combination I/O chip 136. An address latch 150 is connected to the EISA bus E and controlled by the combination I/O chip 136 to provide additional addressing capability for the NVRAM 148. Preferably the NVRAM 148 is used to contain certain system information. A data buffer 152 is connected to the SD portion of the EISA bus E to provide an additional data bus XD for various additional components of the computer system. The NVRAM 148 is connected to the XD data bus to receive its data bits. A flash ROM 154 receives its control and address signals from the EISA bus E and is connected to the XD bus for data transfer. Preferably the flash ROM 154 contains the BIOS information for the computer system and can be reprogrammed to allow for revisions of the BIOS. An 8742 or keyboard controller 156 is connected to the XD bus and EISA address and control portions 108 and 112. The keyboard controller 156 is of conventional design and is connected in turn to a keyboard connector 158 and a mouse or pointing device connector 160.

The computer system of the preferred embodiment also includes audio capabilities. To this end a CODEC chip 162 is connected to the miscellaneous system logic chip 132 and to an analog amplifier and mixer chip 164. An FM synthesizer chip 166 is connected to the analog

amplifier and mixer 164 and receives digital information from the XD bus.

The FM synthesizer 166 is also connected to the control and data portions 110 and 112 of the EISA bus E and is controlled by the miscellaneous system logic chip 132. An audio connector 168 is provided to allow external audio connections to the computer and is connected to the outputs and inputs of the analog amplifier and mixer 164.

Referring now to Figure 2, the processor board P is shown. In the processor board P of Figure 2, the CPU or processor 200 can be any of a plurality of processors, such as the 486DX/33, 486DX2/66, 486DX4/50-100, 486DX4/33-100, 486DX4/33-83, P24T, Pentium 50/75, Pentium 60/90, and Pentium 66/100, and other similar and compatible processors. The processor 200 provides data, address and control portions 202, 204 and 206 to form a processor bus PB. A level 2 (L2) or external cache memory system 208 is connected to the processor bus PB to provide additional caching capabilities to improve performance of the computer system. The L2 cache 208 can be organized as a 128 kbyte direct mapped cache or 256 kbyte two-way set associative cache when used with 486 family processor and as a 256 or 512 kbyte direct mapped or two-way set associative cache when used with Pentium family processors. A cache and memory controller (CMC) and PCI bridge chip 210, is connected to the control portion 206 and to the address portion 204. The CMC 210 is connected to the L2 cache 208 as it incorporates the cache controller and therefore controls the operations of the cache memory devices in the L2 cache 208. The CMC 210 is also connected to control a series of address and data buffers 212. The data buffers 212 are utilized to handle memory data to a main memory array 214. The data buffers 212 are connected to the processor data portion 202 and receive control signals from the CMC 210. The data buffers 212 provide a memory address bus 216 and a memory data bus 218 to the memory array 214. A memory control signal bus 220 and memory address

bus 216 is provided from the CMC 210. Clock distribution and generation circuitry 222 is associated with the processor card P and is connected to the CMC 210. A processor connector 224, such as a card edge, is provided to be matedly received by the processor connector 114. The processor connector 224 is connected to the CMC 210, the data buffers 212 and the clock distribution circuitry 222 to provide clocks to the computer system and to provide a PCI interface to allow the processor 200 to access the PCI and EISA buses P and E and to allow PCI and EISA bus masters to access the main memory array 214.

Referring now to Figure 3, an exemplary video graphics adapter is shown. A video graphics controller 300 is connected to a PCI graphics connector 316, which is matedly received in the graphics connector 116. A ROM 302 is connected to the graphics connector 316 and receives control signals from the video graphics controller 300. Video memory 304 is used to store the graphics data and is connected to the video graphics controller 300 and a digital/analog converter (DAC) 306. The video graphics controller 300 controls the operation of the video memory 304, allowing data to be written and retrieved as required. A video connector 308 is connected to the DAC 306. A monitor (not shown) is connected to the video connector 308.

It is noted that the above description of a computer system is provided for completeness and numerous variations could be developed as apparent to those skilled in the art.

Referring now to Figure 4, a block diagram of the memory controller 210 is shown. There are three main control blocks in the memory controller 210 and three primary address transfer queues. The three primary control blocks are the processor control or PCON block 400, the PCI control or ICON block 402, and the memory control or MCON block 404. The PCON block 400 provides the interface to the processor bus PB,

particularly the processor address bus 204 and the processor control bus 206. Additionally, the PCON block 400 is connected to the L2 data cache 208 and provides the necessary control signals. The L2 cache controller is contained in the PCON block 400. In addition, the PCON 400 provides signals to control the data buffers 212. The ICON block 402 provides data buffer control signals to the data buffer 212 and in addition interfaces to the PCI bus 98, particularly, the control and address portions. The MCON block 404 is the memory device control portion and is connected to the memory address bus 216 and the memory control bus 220, and additionally provides signals to control the data buffers 212. Each of the particular control blocks 400, 402 and 404 control different portions of the data buffers 212 as will be illustrated.

The memory controller 210 and data buffer 212 are effectively organized as a triangle or delta so that data transfer can occur between the processor bus PB and the PCI bus 98, between the processor bus PB and the memory bus, and between the PCI bus 98 and the memory bus. To this end the PCON block 400 is connected directly to the ICON block 402 to provide read addresses for read operations by the processor 200 to the PCI bus 98. The write addresses are provided from the PCON 400 to a P2I queue 406. Preferably, the P2I queue is four operations deep, so that four write operations can be posted to the queue and thus be pending at one time. It is noted that in the case of a 64 bit Pentium microprocessor 200, this would translate to 8, 32 bit PCI bus 98 operations. The output of the P2I queue 406 is provided to the ICON block 402 to provide an address stream. In similar fashion, the read address information and address bank information is provided from the PCON block 400 to the MCON block 404 for processor reads from the main memory 214. The write addresses and bank information are provided from the PCON block 400 to a P2M queue 408, which is again also

preferably four quadword addresses deep, so that four write operations from the processor 200 can be posted to the memory array 214 if a Pentium processor and eight operations if a 486-based processor. The output of the P2M queue 408 is provided to the MCON block 404.

The ICON block 402 is connected to the MCON block 404 to provide read addresses and memory bank information for read operations from the PCI bus 98 to the memory 214. In addition, the write address information and bank information is provided from the ICON block 402 to an I2M queue 410. Preferably, the I2M queue 410 is capable of posting eight quadword addresses to provide relatively deep posting from the PCI bus 98 to the memory 214. The output of the I2M queue 410 is connected to the MCON block 404. An SMAP or system map block 412 is connected to the control block 400, 402 and 404 to provide various signals as necessary.

In addition, a signal referred to as M2LABORT is provided from the ICON block 402 to the MCON block 404 to allow the MCON block 404 to determine when a Memory Read Multiple operation has been aborted by the PCI bus master, as well as for other PCI abort operations. For this discussion, only the Memory Read Multiple case is of interest and the other cases will not be discussed. The MCON block 404 provides a snoop request or SNPREQ signal, the M2l read address that is the address for memory to PCI read operations and the I2M write address to the PCON block 400. This allows the PCON block 400 to perform snoop operations with the L2 cache controller and to provide the operation to the processor 200 so that the L1 cache controller inside the processor 200 can also perform a snoop operation. Snooping of read addresses as well as write addresses is necessary because the L2 cache controller, and the L1 cache controller in the processor 200 in certain cases, are preferably organized as writeback cache controllers, and therefore, snoop operations must occur on reads to maintain memory coherency. The PCON

block 400 provides the SNPDONE and SNPHITM or snoop done and snoop hit to modified signals to the MCON block 404 to allow the MCON block 404 to proceed with the read or write operations or retry a read operation if appropriate.

Referring now to Figure 5, more details of the data buffers 212 are shown. It can be noted that the data buffers 212 also contain queues 411, 413 and 414 similar to those in the memory controller 210 so that addresses are tracked in the memory controller 210 while the data is maintained and transferred through the data buffers 212. The memory controller 210 is organized to control the data buffers 212 such that the particular control blocks 400, 402 and 404 control their respective portions of the data buffers 212. For example, the PCON block 400 controls the latching of data from the processor data bus into the various registers and the output enable or driving of the data from the memory onto the processor data bus 202. Similarly, the ICON block 402 handles latching of data into the various registers from the PCI bus 98 and the output enables of the data to the PCI bus 98. The MCON block 404 handles the operation of shifting data through the various queues 411, 413 and 414, driving data from the queues 411, 413, 414 to the actual memory devices and latching data as required from the memory devices into either the I2M queue 414 or to registers as provided to the processor data bus 202 and the PCI bus 98. It is noted that the processor to memory and processor to PCI queues 411 and 413 are unidirectional in the data buffers 212 but the PCI to memory queue 414 is operated bidirectionally, that it is used for both write data and the read ahead data. Operation will be clearer according to the description below.

One of the features of the memory controller 210 of the preferred embodiment is that it allows great flexibility in the use of various

speed microprocessors and various speed DRAM devices in the memory array 214. Indeed, it allows the memory devices to vary from bank to bank, the main memory array 214 preferably being formed of eight banks or modules. The memory controller 210 of the preferred embodiment allows three different memory speeds to be utilized, with each bank being designated as one of the particular speeds. Each particular memory speed then has a series of parameters relating to timing values as commonly utilized in DRAM specifications. For each particular memory speed, a value to indicate the row address hold time from RAS* is provided, the column address set up times to the CAS* strobe for both reads and writes are provided, the row address set up time to the RAS* signal is provided, as is the memory data set up time to the CAS* strobe and the pulse width of the CAS* strobe. In addition, the RAS precharge time is provided for each memory speed. An indication of when the cycle ready signal is provided to the processor 200 or PCI bus 98 is also indicated as a timing parameter. Preferably, these timing parameters utilize 10 bits of memory in a 16 bit register, the organization shown in Table 1.

TABLE 1

Memory Timing Control Register

Bit 15 Reserved

Bits [14:13] RAS precharge

00 = 2 clocks

01 = 3 clocks

10 = 4 clocks

11 = 5 clocks

Bits [12:11] MRDY read state

00 = Assert MRDY in the C1 state

01 = Assert MRDY in the C2 state

10 = Assert MRDY in the C3 state

11 = reserved

Bits [10:9] reserved

Bit 8 CAS pulsewidth

0 = 2 clocks

1 = 3 clocks

Bits [7:5] Reserved

Bit 4 MD setup to CAS

0 = 1 clock setup time

1 = 2 clocks setup time

Bit 3 Row Address Setup to RAS

0 = 1 clock setup time

1 = 2 clocks setup time

Bit 2 Column Address Setup to CAS, writes

0 = 1 clocks setup time

1 = 2 clocks setup time

Bit 1 Column Address Setup to CAS, reads

0 = 1 clocks setup time

1 = 2 clocks setup time

Bit 0 Row Address Hold from RAS

0 = 1 clock hold time

1 = 2 clocks hold time

RAS Precharge defines the number of clocks in which all RAS* signals must be negated (high) before any are allowed to be asserted (low) again.

MRDY read state defines which CAS state MRDY should be asserted for a memory data read. If the CAS pulsewidth is two clocks wide, C2 is skipped. The MRDY state varies on clock frequency and memory speed.

CAS pulsewidth defines how many clocks CAS* signals must be held asserted.

MD Setup Time to CAS defines how many clocks the MD bus must be driven from the data buffers 212 before CAS* is asserted for a write.

Column Address Setup to CAS, writes defines the number of clocks after the Column Address is generated before CAS* can be asserted for a write cycle.

Column Address Setup to CAS, reads defines the number of clocks after the Column Address is generated before CAS* can be asserted for a read cycle.

Row Address Hold from RAS defines the number of clocks after RAS* asserts before the Row Address may change.

Referring then to Figure 6, each memory bank includes a related register 500, 502 and 504 to indicate the ending address of the particular memory bank. These addresses can be written by the processor 200 after determining the particular size of the memory module located in the particular bank. The output of the bank end address registers 500, 502 and 504 are then provided as one input to comparators 506, 508 and 510. A second input to the comparators 506, 508 and 510 is provided by the memory address of the particular module such as the processor bus PB or PCI bus 98 as appropriate. It is noted that only three of the eight banks are shown for simplicity. The comparators 506, 508 and 510 have two outputs and a third input. One output is the less than output, so that if the provided memory address is less than the ending address of the particular bank, and there is a carry in value to indicate enabling of the comparator, the less than value is true or high, indicating that this is the selected bank. The comparator second output is a greater than or equal to signal, which is used to enable the next comparator. For

example, comparator 506 has its greater than or equal to output connected to an enable input of the comparator 508 such that if the memory address is greater than or equal to the address provided in the bank 0 end address register 500, the comparator 508 is activated. If the address is then less than the bank 1 ending address as stored in the register 502, then the output of the BANK 1 signal of the comparator 508 is activated. If the memory address is instead higher, then this chain continues to determine if the memory address is less than that contained in any of the banks. Therefore, there are eight particular bank outputs, BANK 0 to BANK 7, to indicate the presence of the memory address in the particular memory bank.

These BANK 0 to BANK 7 signals are provided to a memory bank value encoder 512, which provides the BANK<2:0> signals, which are stored along with the write address values in the 12M queue 410 or P2M queue 408, for provision to the MCON block 404.

As noted above, each bank could be one of three particular memory speeds, and these indications are contained in registers 514 and 516. Only two registers are shown for simplicity. The outputs of the registers 514 and 516 are two bits to indicate the particular speed, and these bits are provided to the inputs of an eight input, 2 bit multiplexor 518 whose select input is connected to the BANK <2:0> signals. The BANK <2:0> signals are provided with the particular address, be it read or write addresses for the PCON block 400 or the ICON block 402, to the multiplexor 518 so that the output of the multiplexor 518 is the particular two bits that indicate the bank for the particular address for which the MCON block 404 is about to execute the memory operation. The output of the multiplexor 518 is connected to the select input of a 3 input multiplexor 520. The multiplexor 520 is connected to the registers 522, 524 and 526 which contain the memory speed timing values for speed

0, 1 and 2, respectively. These memory timing values are those indicated in Table I. The output of the multiplexer 520 is timing values as utilized in the state machines and logic described below for the MCON block 404. In addition, the address values $\langle 27..3 \rangle$ and byte enable or BE $\langle 7..0 \rangle$ signals are also provided to the MCON block 404 as they have been tracked through the queues 408 or 410 or are provided directly on read operations. Therefore, the memory bank speed timing values are directly provided by first decoding the particular bank, then having a look up for the bank based on the previously defined and stored memory speed value of the optional memory speed settings, and then selecting the particular memory speed values for operation for that particular memory cycle.

An example of the timing values for the preferred embodiment for three processor speeds and two memory device speeds is shown in Table II.

TABLE II
Memory Timing Control Combinations

Parameter	33MHz		50MHz		60MHz	
	60ns	70ns	60ns	70ns	60ns	70ns
RAS precharge	2	2	3	3	3	4
MADY assertion state	C1	C1	C3	C3	C2	C3
CAS pulsewidth	2	2	2	2	2	3
MD Setup	1	1	1	1	2	2
Write Column Addr Setup	1	1	1	1	1	1
Read Column Addr Setup	1	1	1	1	1	1
RA hold	1	1	1	2	2	1
Memory Timing Control Register (hex value)	00__00	00__00	30__00	30__01	20__11	51__10

The DRAM device speeds can be determined by reading encoded bits conventionally provided on SIMMs and checking the size. These two parameters are then used in a lookup table to determine SIMM speed. This technique is described more fully in Serial No. 08/034,105, filed March 22, 1993, which is hereby incorporated by reference. With the DRAM device speed then known for each bank and microprocessor speed known by other encoded bits, these two parameters are then used in a second lookup table to obtain the binary value to be placed in the timing value register for that combination. The computer system designer knows the various propagation delays of the various paths and can then simply use the clock speed and memory timing parameters provided by the memory device manufacturer to determine the timing values for each combination.

Proceeding then to Figure 7, the use of the addresses, byte enables and timing values are shown as being provided to a portion of the MCON block 404. The processor write request, processor read request, refresher request, PCI read request and PCI write request signals are inputs to an arbiter 550 along with signals referred to as I2CAMHIT, I2MQ_EMPTY and P2MQ_EMPTY. The various request signals are an indication that operations are pending for the memory controller to operate on from the indicated sources of the indicated type. Development of these signals is not fully described but can be readily developed by one skilled in the art. The outputs of the arbiter 550 are the P2M, M2P, I2M and M2I signals to indicate which source and direction of operation has occurred, that is P2M for a processor to memory write, M2P for a processor to memory read, I2M for a PCI to memory write operation and M2I for a PCI read operations. These four signals are provided as the select signals to a four input multiplexor 552 which receives the particular read or write addresses, the byte enables, the memory timing values and the bank indications for the particular source. The P2M queue

408 provides write addresses, byte enables, memory timings and bank information from the queue 408 for processor to memory write operations, while the read addresses, byte enables, memory timings and bank information are provided directly for processor reads in the M2P case. Similarly, the I2M queue 410 provides write addresses, byte enables, memory timings and bank information from the I2M queue 410, while the read addressing information is provided directly from the ICON block 402 to the MCON block 404. The output of the multiplexor 552 is the particular address value, the byte enables the bank value to indicate which particular bank and the memory timing values for the particular memory operation to be performed by the memory controller 210 to access the necessary memory location for either a read or a write as appropriate. It is also noted that a page hit detector 554 is connected to the output of the multiplexor 552 to provide a PAGEHIT signal to allow the memory controller 210 to determine whether a page hit has occurred, so that it can operate the DRAM devices in the desired page mode operation for best performance.

Proceeding then to Figure 8, blocks representing various state machines and other logic utilized in the MCON block 404 are shown. A number of state machines provide the actual information and control functions and are interlocked. The primary state machine is the MEMSM or memory state machine 600. The memory state machine 600 receives the memory timing values from the multiplexor 552 and the M2IABORT signal from the ICON block 402. A precharge state machine 602 is provided to provide a signal referred to as RASPCHG or RAS precharge to indicate that the precharge time for the particular memory bank is completed if page mode operation is not occurring. The RASPCHG signal is provided to the memory state machine 600. Additionally, RASUP or RAS up logic 604 provides a RAS__UP signal which is utilized by the memory state

machine 600 to determine whether to leave the RAS* or row address strobe signal high or low when no memory cycles are pending, thus providing a prediction whether the next operation will be a page hit or page miss to thereby improve performance. Detailed operation of this logic will be described below.

A refresh state machine 606 is provided to control refresh operations but will not be described in detail in this description because its operation is relatively conventional and is omitted for clarity. A memory address state machine 608 receives the timing values from the multiplexor 552, as well as the basic clock signal of the system and provides interlock signals to the memory state machine 600, the MSELRA or memory select row address signal and the MWE* or memory write enable signal. The MSELRA signal is provided to the select input of a 2:1 multiplexor 610 which receives at its inputs the memory addresses appropriately connected to provide row and column addresses based on the selection input. A burst address block 612 is provided and connected to the column addresses to simplify burst operation. The outputs of the multiplexor 610 are the memory addresses provided to the memory array 214 over the memory address bus 216 for the particular operation. The MWE* signal is similarly the memory write enable signal as provided to the main memory array 214. A memory data control state machine 614 is provided. It receives certain of the memory timing values and provides interlock signals to the memory state machine 600 and controls the operation of pertinent portions of the data buffers 212.

The memory state machine 600 provides the MARS or master RAS and MCAS or master CAS signals. The MARS signal is combined with the bank value signals for the particular memory operation to provide the RAS* <7:0> signals which are provided to the particular banks. If the bank value indicates that this is the particular bank value for the

operation, then the RAS logic 616 directs the MARS signal to that particular bank in an inverted form. If the bank value is not the one particularly being addressed, then the RAS* signal for that particular bank is held high to allow the memory devices to be fully precharged. The MCAS signal is provided to CAS logic 618 which also receives the BE* <7:0> signals for the preferred 64 bit width. These signals provide the eight, byte lanes of information that are encoded with the MCAS signal to produce the CAS* <7:0> signals which are used to enable the particular memory devices of the bank indicated by the RAS* <7:0> signals. If a particular byte enable is not activated, then its related CAS* signal is not activated, and therefore the memory device is not selected as it does not receive a column address. This allows the use of a single MWE* signal, with the CAS* signal providing the byte lane encoding.

The queues 406, 408 and 410 are organized as a series of registers of the appropriate width. Particular registers in each queue are selected based on three particular counter values associated with each queue. The three counters are the read location counter, the write location counter and the data contents counter. The read location counter is utilized to determine the particular location of the four or eight registers from which the read data for the next cycle is to be obtained and provided to the particular block. The write location counter is utilized to indicate the particular register into which data is to be written in the next operation. The read and write location counters increment on each read and write operation, respectively, and thus operate in a circular fashion. The data contents counter is utilized to indicate whether there is actually data in the queue. The data contents counter is an up/down counter. The data contents counter counts up on a write operation to the queue and decrements on read operations. If the data contents counter indicates that the queue is full by being

at a maximum value, then data is not written into the queue until data has been removed from the queue. These counters are conventional logic and are not shown for simplicity and clarity. I2MQ_EMPTY and P2MQ_EMPTY signals thus correspond to the data contents counters being at a zero value.

Because of the length of the I2M queue 410, it is organized as a content addressable memory or CAM. I2M queue registers 640 and 642 are shown in Figure 9, with many omitted for simplicity. The output of the particular register 640, 642 is not only provided to the MCON block 404 via a multiplexer but is also provided to an associated comparator 644 and 646. The second input of the comparator 644 and 646 receives the processor address being provided on the processor address bus 204. The comparators 644 and 646 have equal outputs so that if the address being provided directly on the processor address bus 204 is equal to one of the addresses contained in the I2M queue 410, then the particular comparator provides its equal signal true. The equal signal output goes to one input of an AND gate 648 and 650. The second input to the AND gate 648 and 650 is provided by a valid bit register 652 and 654. If data is not contained in the particular register 640, 642 in the I2M queue 410, then the valid bit register is set to 0, whereas if data is present, then the bit is set to 1. These valid bit registers are set when data is written into the I2M queue 410 and cleared when data is read out of the queue 410. In this manner, if the register is valid and the comparator determines that there is an equality, the output of the particular AND gate is set to a one value to indicate that a hit has occurred. The outputs of the AND gates 648 to 650 are provided to an eight input OR gate 654 whose output is the I2MCAMHIT signal, which is provided to indicate that the particular read address being requested by the processor is actually present in the I2M queue 410 as a write operation

from the PCI bus 98, and therefore the memory read request from the processor 200 cannot occur until the particular location has been flushed out of the I2M queue 410. Operation of this delay is seen in the arbiter 550 as described below.

In the preferred embodiment, there are five possible requests for the main memory 214: a processor-to-memory write (P2M), a memory-to-processor read (P), a PCI-to-memory write (I2M), a memory-to-PCI read (M2I), and refresh. A P2M write refers to a write to the memory 214 and a M2P read refers to a read of the memory 214, both initiated by the microprocessor 200. An I2M write refers to a write to the memory 214 and a M2I read refers to a read of the memory 214, both initiated by a PCI bus master. All memory requests from EISA bus masters are passed through the PCI-EISA bridge 130, which includes the necessary logic to be a bus master on the PCI bus 98. Thus, any EISA originated memory request is effectively a memory request asserted by the PCI-EISA bridge 130.

Generally, the priority of the memory requests areas follows, with some exceptions: (1) second refresh request; (2) P2M write request; (3) M2P read request; (4) I2M write request; (5) M2I read request; and (6) first refresh request. The second refresh request indicates that two refreshes are outstanding. When that occurs, the memory controller 234 gives both outstanding refresh requests the highest priority, executing both refresh cycles. The P2M write request is always higher in priority than other memory requests except the second refresh. However, if the I2M queue 410 is full or the I2M queue 410 is not empty and a PCI bus 98 read is outstanding, a signal M__P2M __NOPOST is asserted to prevent further queuing of P2M write requests until a PCI write operation is completed. This allows the P2M queue to clear out, thereby allowing requests from the PCI bus 98 to be serviced. However, assertion of the signal M__P2M __NOPOST does not prevent writeback cycles from being

queued, as the writeback may be needed by the PCI memory request.

The M2P read request is always lower in priority than the P2M write request, but it is usually higher in priority than I2M write and M2I read requests. The operation can only occur when the P2M queue 408 is empty. However, an unlocked M2P read request is forced lower in priority than an I2M write request if the M2P read is to the same address as an I2M write pending in the I2M queue. When this occurs, the M2P request remains lower in priority than I2M requests until the I2M write request having the matching address is written to the main memory 214. A M2P read request is also forced lower in priority than an I2M write request if the I2M queue is full. Additionally, if an M2I read request is asserted while an I2M write request is pending, the I2M write request is forced higher in priority than the M2P read request to allow the I2M queue to clear, thereby allowing the M2I request to proceed. Further, an M2I read request is forced higher in priority than the M2P read request if the M2I read has been waiting for the M2P request to negate for more than one arbitration cycle.

The I2M write request is always lower in priority than the second refresh request, the P2M write request, and it is generally lower in priority than the M2P read request with the exceptions noted above. The I2M write operation can only occur when the P2M queue 408 is empty. The I2M write request is always higher in priority than the M2I read request. The I2M write request is held off if the processor is performing a locked access of the main memory 214. Thus, for a locked processor cycle, the exceptions discussed above do not apply to override the higher priority of M2P read requests over I2M or M2I requests.

A locked or atomic access of the main memory 214 is indicated by a signal LOCK* driven by the microprocessor 200. A locked cycle allows the microprocessor 200 to read an address location in the main memory 21

4 and be assured that the accessed location is not changed by another bus master before the microprocessor 200 writes back to the same memory location. These type cycles are referred to as read modify write cycles.

Locked cycles are also generated during other bus transfers, such as during execution of the XCHG (exchange) instruction when one of its operands is memory-based, when updating a segment or page table entry, and when executing interrupt acknowledge cycles.

The M2I read request is always lower in priority than the second refresh request, the P2M write request, and the I2M write request. However, it is higher in priority than the unlocked M2P read request in the instance noted above. The M2I read operation can only occur when the P2M queue 408 is empty.

Finally, the first refresh request is always lower in priority than any of the other requests. However, as noted above, when the second refresh request is asserted, both the first and second refresh requests are executed regardless of whether other requests are pending.

Further details of this arbitration are provided in Serial No. 08/324,011, entitled "System for Arbitrating Access to Memory" filed concurrently herewith, which is hereby incorporated by reference.

Referring now to Figure 10 the memory state machine 600 is shown in detail. Indications in italics next to a particular state bubble indicate the value of various output signals, developed by the state machine. In this case the MARS and MCAS signals. The phrases adjacent to an arrow going from one bubble to another are the logical conditions on which that path is taken. If a path does not have an adjacent logical condition, that path is taken in all other cases. Transitions are made on the rising edge of the processor clock signal. Upon reset of the computer, operation of the state machine 600 transfers to the R1 or RAS

idle state. In this state the MARS signal is not asserted if the next state is not state R1. If the next state is R1, then the MARS signal is set to the value 1 or asserted so that the row address strobe signal is provided. The MCAS signal is negated to a zero level. Control proceeds from state R1 to state R1 if the M2P condition is true so that a processor read operation is occurring, if a PCI read operation is occurring (M2I) which has not been aborted (!M2IABORT) and is not being cleared (!CLR12M), or if a write operation from the processor or the PCI bus is occurring; the row address ready signal (RADRRDY) is true, which indicates that the row address set up time to the RAS signal has been met as indicated by the memory address state machine 608, or, if in a processor read case there is no wait signal indicated as would be appropriate if an aborted memory to PCI read ahead is completing (WALTRADRRDY) and the row address set up time two (RADSETUP2) signal is not activated which indicates that relatively fast memory is present; a refresh cycle is not occurring (RFSH_ON); the current read cycle to the PCI bus 98 is not being aborted (M2IABORTCUR); and the precharge period has completed for as indicated by the RASPCHG signal. The M2IABORT signal is provided by the ICON block 402 as described above. The M2IABORTCUR signal indicates that the cycle currently being performed by the MCON block 404 is to be aborted or terminated as soon as possible. The M2IABORTNEXT signal indicates that the next M2I cycle which is pending is to be aborted, which occurs by simply skipping the cycle. The development of the CLR12M signal is detailed below, but briefly indicates that a snoop read hit to a modified location has occurred, so that the 12M queue 414 must be flushed and the read cycle reexecuted. Therefore, if a memory cycle is indicated, the row address set up time has been met, and the device is precharged, control proceeds in state R1 to state R1. In all other cases control remains at state R1.

In state R1 the MARS signal is set to 1 or high and true, and the MCAS signal is set low to indicate that this is a RAS only portion of the cycle. Control proceeds from state R1 to state R2 if the RADHLD2 signal is true, which indicates that two clock periods of hold time are required and therefore the intermediate R2 state is required. If the RADHLD2 signal is not true, then only 1 clock of hold time is required from the activation of the RAS* signal and control proceeds directly from state R1 to state R3. In state R2, the MARS signal is asserted and the MCAS signal is negated. Control proceeds from state R2 to state R3 on the next rising edge of the processor clock signal.

In state R3, the MARS signal is true or high, while the MCAS signal is set high if the next state is to be state C1 and otherwise the MCAS signal is set low. Control proceeds from state R3 to state C1, the first of the column states, if the column address ready (CADRRDY) signal is true as provided by the memory address state machine 608 to indicate that the particular column address set up time, be it read or write, has been satisfied and either a PCI read operation is occurring or a memory to processor bus read operation is occurring, and the memory operation is not stalled waiting for read data; or a write operation from the processor to memory or the PCI bus to memory is occurring with the data being valid as indicated by the MDVALID signal which is provided by the memory data state machine 614. Therefore, if the column address set up time has been met and indication of write operation data being properly valid for write cases is provided, control proceeds from state R3 to state C1. Otherwise, control remains in state R3 waiting for the timing conditions to be satisfied.

In state C1, both the MARS and MCAS signals are high to indicate that the CAS portion of the cycle is occurring. Control proceeds from state C1 to state C2 if the CASPW3 or CAS pulse width 3 clock signal is

true. If the CASPW3 signal is not true, control proceeds directly from state C1 to state C3, with control always proceeding from state C2 to state C3 on the next rising edge of the processor clock signal. In state C2, the MARS and MCAS signals are both also true. Thus, state C2 is skipped if the CAS pulse width can be narrower, as would be true for faster memory or slower processors with an additional clock period provided when necessary.

In state C3 the MCAS signal is negated to a 0 level to indicate completion of the particular memory cycle and the MARS signal is set to 0 if the next state is the R1 state or if the particular cycle is aborting. Otherwise, the MARS signal remains in a high or asserted state to allow operation in page mode if appropriate. Control proceeds from state C3 to state R1 if a burst operation is not occurring, which is a preferable operation for 486 and Pentium processors, or the current memory to PCI read cycle is being aborted as explained below; the next memory to PCI read operation is not aborted; there is no indication that the M2I cycle is to be aborted as indicated by the M2IABORT signal or the CLR12M signal; and a non-refresh cycle is occurring (M2P+ADDRDY) and this is not a page hit; a refresh cycle is occurring or there is no cycle pending and the RAS__UP signal is true, indicating that the RASUP block 604 has indicated that the prediction is for the next cycle to be a page miss. The RAS__UP signal is generated as follows:

$$\begin{aligned} \text{RAS_UP} = & ! ((\text{!WRITEBACK} \ \&\& \ \text{P2M}) \ | \ | \ (\text{!MDATARD} \ \&\& \ \text{M2P}) \ | \ | \\ & (\text{!Q_I2MQEMPTY} \ \&\& \ \text{I2M}) \ | \ | \ (\text{!M2IABORTCUR} \ \&\& \\ & \text{M2IMURFAD} \ \&\& \ \text{!M2IABORTNEXT}) \ | \ | \ \text{HIT2MOD}) \end{aligned}$$

Therefore, the RAS+ signal is kept low or in page mode following processor to memory write operations which were not writebacks from the

L1 or L2 caches, processor code read operations, PCI bus write operations with pending write operations, Memory Read Multiple operations from the PCI bus 98 which are not being aborted, or bits to a modified address, indicating that a writeback operation has just occurred. In all other cases the RAS_{UP} signal is true, such that it is considered better to initiate a full new page access to allow precharging to occur during the idle period.

If the conditions for transfer from state C3 to R1 are not true, control proceeds from state C3 to state RN or the RAS low idle state. In this state the MARS signal is low if the next state is state R1 otherwise the MARS signal is asserted. The MCAS signal is asserted if the next state is C1, otherwise the MCAS signal remains in low state. Control proceeds from the RN to C1 state for burst operations in general if the column address set up time has been met and there are no abort operations occurring. Control proceeds from the RN state to the R1 state under conditions which generally indicate that the cycle is being aborted or a processor non-burst read which is a page miss is next to occur. In all other cases, control remains at state RN, idling with the RAS_{UP} signal low until a cycle needs to be run or an abort or page miss occurs. Therefore, it can be seen that the memory state machine 600 drives the MARS and MCAS signals based on the timing values of the proper bank, these timing values being variable as indicated above.

A memory address state machine 608 is shown in Figure 11. The state machine 608 starts at state RAI upon reset. In this state, the RADRRDY signal and CADRRDY signals are set low to indicate that the row and column addresses are not ready, and the MSELRA signal is set high to initially select the row address to be provided to the memory devices. The MWE_{UP} signal is set to 1 or negated level on reset and after entry into the RAI state and is set equal to the CH1 state value the first

state after entry from the CH1 state as described below. Control proceeds from the RA1 state to the RH1 state if an active cycle is pending from the PCI bus 96 or processor 200 and the next cycle in the read ahead operation is not to be aborted. The current M2I read cycle is aborted under certain conditions, such as receipt of M2IABORT signal or a writeback is occurring, while the next M2I read cycle is aborted when the M2INA signal has been received but the cycle not started when the M2IABORT signal is received. In all other cases control loops at state RA1.

In state RH1 the CADRRDY signal is set low and the RADRRDY signal is set to the value of M2P or memory to processor read upon entry from the RA1 state, is equal to 0 on entry from the CH2 state as defined below and after entry into the RH1 state is set to a 1 level. The MSELRA and MWE* signals are set to high levels so that the row address is selected and a write operation is not indicated. Control proceeds from the RH1 to the RH2 state if the MARS signal is true from the memory state machine 600 and the RADHLD2 signal or row address hold time signal is set to indicate slower memory devices. If the MARS signal is true and the RADHLD2 signal is not set, control proceeds from the RH1 state directly to state CH1, which is also where control proceeds on the next clock signal in the RH2 state. In all other cases, operation loops at the RH1 state. In the RH2 state, the RADRRDY signals and CADRRDY signal are set low, and the MSELRA and MWE* signals are set at high state. Control proceeds from the RH2 to the CH1 state.

In the CH1 state, the RADRRDY signal is low to indicate that the row address set up time has not been met. The CADRRDY signal is set to a

In state RH1 the CADRRDY signal is set low and the RADRRDY signal high or true value after initial entry into the state. On entry the CADRRDY signal is set high if short CAS address setup times were set

(!CADSETUP2) for the read or write as appropriate, and otherwise is set low. The MSELRA signal is set to 0 to indicate that the column addresses are selected and the MWE* signal is low if a write operation is occurring.

Control proceeds from the CH1 state to the RAI state if the memory state machine 600 is in state RN and either the cycle is aborting or a processor read page miss is pending. Control proceeds from the state CH1 to state CH2 if the memory state machine is not in state RN or the cycle is not aborting completely or the processor read cycle is a page hit. In the CH2 state the RADRRDY and CADRRDY signals are set low to indicate that no addresses are ready. The MSELRA signal is set low and the MWE* is set low to indicate the write operation is occurring and the column addresses are provided. Control proceeds from the CH2 state to the CH1 state for ongoing burst cycles when the memory state machine is in state C3 or for pending processor read page hit operation. Control proceeds from the CH2 state to the RAI state if the cycle is aborting or if an idle condition is present and the RAS* signal is to be set high. Control proceeds from the CH2 to the RU1 state if a processor read page miss operation is to occur and the memory state machine 600 is at the end of a cycle.

Therefore the memory address state machine 600 properly provides the MWE* signal for the DRAM devices and controls the address multiplexer 610. In addition, the RADRRDY and CADRRDY or row and column address ready signals are provided to the memory state machine 600 based upon the column and row address set up times as provided in the memory timing registers.

The MD or memory data state machine 614 is illustrated in Figure 12. The state machine 614 starts at the IDLE state upon reset. Control

proceeds from the IDLE state to a RD state for processor or PCI bus 98 read operations which are not being aborted and refresh is not occurring. Control proceeds to the DCD state if a processor to memory write operation is occurring and there is no refresh or if a PCI bus to memory write operation is occurring and again there is no refresh. Control otherwise remains at the IDLE state. Control proceeds from the RD state to the DCD state if the memory state machine is in states C3 or RN, the cycle is either being aborted or is not a burst cycle and it is a write operation with a 33 MHz processor. Control proceeds from the RD state to the IDLE state if MARS is negated and the cycle is aborting or the memory state machine is in states C3 or RN, the cycle was not a burst or is aborting, and a non-33 MHz processor write operation or PCI bus 98 write cycle is pending or no cycles are active. In all other cases, control remains at state RD.

Control proceeds from the DCD state to the DH1 state if the MDSETUP2 signal is not true, indicating that this is a fast memory data set up case, and the memory state machine 600 is not in state C2 and 33 Mhz operation is indicated for the processor. In all other cases, control proceeds from the DCD state to the DS2 state. Control proceeds from the DS2 state to a DS3 state if the processor 200 is not operating at 33 Mhz and the MDSETUP2 signal is set to indicate slower memory data set up times. Control proceeds from the DS2 to DH1 states in all other cases. Control proceeds from the DS3 state to the DS1 state on the next rising edge of the clock signal.

In the DH1 state, the MDVALID signal is set to 1 or asserted to indicate that the memory data set up time to the column address strobe has been met. This allows the memory controller state machine 600 to proceed to state C1. Control proceeds from the DH1 state to the DCD state if further write operations are pending and otherwise proceeds to

the DH2 state. In the DH2 state, the MDVALID signal is set to 0 to indicate that the memory is no longer valid. The value of the MDVALID signal is not changed in any states other than the DH1 and DH2 states, except it is cleared on reset. Control proceeds from the DH2 state to the IDLE state if no write operations are pending from the PCI bus 98 or from the processor 200. In all other cases, control proceeds from the DH2 state to the DCD state. In this manner, the memory data state machine 614 provides the MDVALID signal to the memory state machine 600 when the memory data is properly set up.

The precharge state machine 602 is shown in Figure 13. Upon reset, the state machine 602 operation commences at state A. Control remains in state A when the MARS signal is not asserted. When the MARS signal is asserted, control proceeds to one of states B, C, D, or E, dependent upon the number of clocks defined for the precharge time. Control proceeds to state B for five clocks, to state C for four clocks, to state D for three clocks and to state E for two clocks. Control proceeds from states B to C to D to E, sequentially, when the MARS signal is not asserted. Otherwise, control remains in each particular state. Control then proceeds from state E back to state A when the MARS signal is deasserted. Therefore the precharge state machine 602 leaves state A upon the commencement of a particular memory operation and then does not begin traversing the remaining states until the MARS signal has been negated, so that a precharge period has started. It is noted that this results in the proper precharge time for any given bank, even if banks are being switched in sequential memory operations if refresh operations are ignored. This occurs because, as noted above, when a particular bank is not selected the RAS* signal for that bank is set high so that it is in a precharge period. Thus if the bank is not selected, it has been through at least one full memory cycle of precharge, which is

sufficient to provide the required recharge in all cases. Thus, the precharge time is set for that of the particular memory bank on which the current cycle is operating, so that if back to back cycles occur on that particular memory bank, the memory devices are properly precharged.

However, refresh operations somewhat degrade this operation as it is not known in the preferred embodiment which bank will be requested after a refresh cycle, so to simplify the design of the preferred embodiment, the precharge value for all of the DRAM types is set to the worst case by the initialization or POST software. This reduces performance in certain cases but simplifies the design. A slightly more complicated design would use the worst case value, preferably provided to a predetermined register for only the first precharge operation after a refresh cycle and thereafter operation would revert to the optimal timing for each bank.

Write addresses and data traverse through the I2M queues 410 and 414 based on having posted data and the arbiter 550 providing access to the queues 410 and 414 by the memory system. The write addresses are provided to the PCON block 400 when placed in the I2M queue 410, to allow the PCON block 400 to control snooping of the address by the L1 and L2 cache controllers. Read operations occur in a different fashion than write operations. Read addresses are provided to the MCON block 404 along with an indication that a read request has issued. The read addresses are provided based on a return of an L2 cache line, which is 4 32 bit words for 486 microprocessors and 8 32 bit words for Pentium processors. Therefore, when an M2I read is requested, four or eight 32 bit words are provided, with the ICON block 402 properly providing the data from the read operation to the PCI bus 98 as required. The read request indications are provided by a memory to PCI next address state machine described below. The read request is also provided to the

arbiter 550 for arbitration and to the PCON block 400 to allow L1 and L2 cache snooping with a writeback in case of a hit to a modified line. When the arbiter 550 provides the approval to proceed with the PCI bus 98 read, the MCON block 404 then proceeds to process the information. It is noted that this will have occurred only after the I2M queues 410 and 414 are emptied of any pending write operations so that the I2M data queue 414 provided in the data buffer 212 can be utilized to do a block read operation of the lengths indicated. Therefore, when the I2M queues 410 and 414 are emptied, the read operation commences and is controlled by an M2I state machine as shown in Figure 14. The state machine commences at state A upon reset. Control then proceeds to state B if the I2M queue 414 is empty, a read is requested and the L2 cache 208 is not being flushed. This would be a normal read condition. If instead, the L2 cache 208 is being flushed, control then proceeds from state A to state F, assuming that the I2M queue 414 is empty and the read request is present. Otherwise control remains at state A waiting for the read operation to be initialized.

From state B, control proceeds to one of four different states. Control returns to state A if the M2IABORT signal has been received, indicating that the PCI bus master has aborted the read operation, or if the IREQACK signal is true, which indicates that the next operation as determined by the arbiter 550 will be a PCI read, and the SNPDONE signal is true indicating that the PCON block 400 has completed snooping the read operation address to the level 2 cache and to the level 1 cache in the processor 200. Control proceeds from state B to state C if the M2IABORT signal is not asserted, the next cycle is a PCI read as indicated by the IREQACK signal being asserted, there has not been a snoop hit to a modified (SNPHITM) location as indicated by the PCON block 400 and the snoop operation has not been completed, as also

indicated by the PCON block 400. This will be the case where a read operation has been requested and has been arbitrated but may have to be aborted because the snoop operation has not been performed and a hit to a modified location may yet occur. Control proceeds from state B to state D, if the operation is not being aborted and a PCI read has not been arbitrated, but there has been a snoop hit to a modified location with the writeback operation not yet completed. control proceeds from state B to state E if the cycle is not aborted, has been arbitrated as indicated by the IREQACK signal and there has also been a snoop hit to a modified location, so that a writeback operation from the appropriate cache controller will be occurring. In all other cases control remains at state B.

Control proceeds from state C back to state A if the cycle is being aborted or if the snoop is completed without being a hit to a modified location. control proceeds from state C to state E if the cycle is not aborted and there has been a snoop hit to a modified location. Otherwise, control remains at state C until the snoop operation is completed, the cycle is aborted or there is a snoop hit to modified. control proceeds from state D back to state A if the operation is aborted or upon indication that the PCI cycle is next for operation and the snoop has completed. This would occur after the writeback has been completed and then the PCI operation is next to occur, as the write back will have superseded the PCI operation. Control proceeds from state D to state E if it is not being aborted, the PCI request is next and the snoop has not been fully completed. Otherwise control remains at state D.

In state E, the CLR12M signal is set to indicate to other portions of the memory controller 210 that because of the writeback, the data in the 12M queue 414 must be flushed and discarded. A CLR12MDONE signal indicates this flushing has been completed. Control proceeds from state

E to state F if the cycle is not being aborted and the signal CLR12MDONE or clear the queue done signal is true and the snoop is not yet completed. Control proceeds from state E to state G if the cycle is not being aborted, the clearing of the queue 414 has been completed and snoop has been performed. Control proceeds from state E to state A if the cycle is being aborted and in all other cases remains at state E.

Control proceeds from state F to state G if it is not being aborted and the snoop cycle has been completed. Other control proceeds from state F back to state A if the cycle is being aborted. Otherwise control remains at state F. Control proceeds from state G back to state A if the cycle is aborted or if it is next in line as indicated by the IREQACK signal.

Therefore, the M2I state machine controls transfer of information from the memory to the PCI interface using the I2M queue 414 located in the buffer 212. Data is not transferred until the queue 414 is cleared of any write data and then proceeds only based on snoop information and when the arbiter 550 allows it to proceed.

As noted above, the design of the preferred embodiment performs read ahead operations when a PCI Memory Read Multiple operation has been received. The operation proceeds as follows. The ICON block 402 receives a Memory Read Multiple cycle from the PCI bus 98 and when there is room in the I2M queue 414 issues an M2I read cycle request, along with an address, to the MCON block 404. The MCON block 404 arbitrates as described above and ultimately starts on the M2I read request. When it has started the operation, it provides a next address or M2INA signal to the ICON block 402 and provides the data to the I2M queue 414 along with the appropriate ready signals. The ICON block 402 knows a Memory Read Multiple command is occurring and issues another M2I read cycle request at the next address when the M2INA signal is received and the I2M queue

414 can receive another cache line of data. The ICON block 402 also receives the ready signals from the MCON block 404 and provides the data to the PCI bus 98. The MCON block 404 receives the M2I read cycle request and executes it when the cycle wins the arbitration. Conventionally this will be before the ICON block 402 has removed all of the data from the I2M queue 414 for the initial read request. The MCON block 404 then commences the read request and issues another M2INA signal. The MCON block 404 then provides the data to the I2M queue 414. The ICON block 402 receives this M2INA signal and again checks to see if there is space available in the I2M queue 414 to receive another cache line. When there is room because the ICON block 402 has provided sufficient data to the PCI bus 98, the next M2I read request is provided to the MCON block 404. This process continues until either the Memory Read Multiple completes, a page boundary is crossed or the PCI bus master aborts the cycle.

The abort case is the one of interest as the pending read ahead operation is terminated as soon as possible to save retrieving the entire cache line. This can be seen in the discussions of the MEMSM 600, the MADRSM 608, the MDCDSM 614 and the M2I state machine. This quick termination is seen in the MEMSM 600 as the return to the R1 or C1 states from the C3 and RN states, so that the cycles finishes as soon as the current individual read operation is completed, thus potentially before the completion of the full cache line read. Similarly the MADRSM 608 returns to the RAI state if a cycle has not started or when the column addresses have been provided. The MDCDSM 614 returns to the IDLE state if no operation has been started or if the MEMSM 600 is in the C3 or RN states. The M2I state machine returns to the A state whenever the M2IABORT signal is received. On detection of the abort, the ICON block 402 determines the end of the read cycle and resets its pointers to the

data in the I2M queue 414 to indicate that no data is present, thus effectively discarding the data which has been read ahead. Thus the read ahead operation terminates as soon as possible after the abort indication is received, saving time by not completing the full cache line retrieval.

The next address indication to allow the read ahead is provided by the M2INA or memory to the PCI next address state machine shown in Figure 15. The initial read address will have been provided using the conventional read request mechanism. An M2INA signal is provided to the MCON block 404 to indicate that the next read cycle can begin. This state machine begins operation in state A upon reset and proceeds to state B if the PCI read operation is next as indicated by the REQACK signal and is not being aborted. Otherwise, control remains at state A. Control proceeds from state B back to state A if the cycle is being aborted or if the I2M queue 414 is to be cleared because of a writeback or if the snoop has been completed and the M2IACK signal is provided indicating that the prior read operation has been acknowledged by the MCON block 404. Otherwise control remains at state B. Control proceeds from state B to state C if the cycle is not being aborted, it is not necessary to clear the queue, the snoop operation has not completed and yet an M2IACK signal has been received. Control proceeds from state C back to state A if the cycle is aborted, the I2M queue 414 is being cleared, or the snoop is completed and otherwise remains in state C. Thus the M2INA state machine returns to idle upon receipt of an abort indication.

The M2INA signal is provided to the MCON block 404 to indicate that the next address is being provided, that is, another read request can be issued to keep the I2M queue 414 filled ahead of the PCI bus 98. The M2INA signal is provided if the cycle is not being aborted, the I2M

data in the I2M queue 414 to indicate that no data is present, thus effectively discarding the data which has been read ahead. Thus the read ahead operation terminates as soon as possible after the abort indication is received. Saving time by not completing the full cache line retrieval.

The next address indication to allow the read ahead is provided by the M2INA or memory to the PCI next address state machine shown in Figure 15. The initial read address will have been provided using the conventional read request mechanism. An M2INA signal is provided to the MCON block 404 to indicate that the next read cycle can begin. This state machine begins operation in state A upon reset and proceeds to state B if the PCI read operation is next as indicated by the IREQACK signal and is not being aborted. Otherwise, control remains at state A. Control proceeds from state B back to state A if the cycle is being aborted or if the I2M queue 414 is to be cleared because of a writeback or if the snoop has been completed and the M2LACK signal is provided indicating that the prior read operation has been acknowledged by the MCON block 404. Otherwise control remains at state B. Control proceeds from state B to state C if the cycle is not being aborted, it is not necessary to clear the queue, the snoop operation has not completed and yet an M2LACK signal has been received. Control proceeds from state C back to state A if the cycle is aborted, the I2M queue 414 is being cleared, or the snoop is completed and otherwise remains in state C. Thus the M2INA state machine returns to idle upon receipt of an abort indication.

The M2INA signal is provided to the MCON block 404 to indicate that the next address is being provided, that is, another read request can be issued to keep the I2M queue 414 filled ahead of the PCI bus 98. The M2INA signal is provided if the cycle is not being aborted, the I2M

queue 414 is not being cleared, the snoop of the previous read cycle has completed and the M2INA state machine is either in state C or in state B and the M2IACK signal has been received. This M2INA signal is an indication that the processing of the prior address is complete by the MCON block 404 and the processing of the next read address can begin. The actual incrementing of the read address value is performed in the MCON block 402 using an 8 bit counter, thus limiting the total read ahead length to 256 address values. When the counter reaches 255, the read ahead operation is terminated by logic not illustrated for simplicity by causing the Memory Read Multiple to be disconnected. A new address must be received from the PCI bus master to continue the Memory Read Multiple Operation.

Therefore a memory controller according to the preferred embodiment includes many improvements. It is very flexible, allowing simple yet high performance use with a plurality of processor and memory device speeds, allowing flexibility and economics of scale. It provides improved performance by basing idle state RAS* signal level predictions on both processor and PCI bus conditions. It allows very deep write points from several buses and easily handles the various coherency issues. It also provides a quick method for ending read ahead cycles when the basic read operation has been aborted.

The foregoing disclosure and description of the invention are illustrative and explanatory thereof, and various changes in the size, shape, materials, components, circuit elements, wiring connections and contacts, as well as in the details of the illustrated circuitry and construction and method of operation may be made without departing from the spirit of the invention.

4. Brief Explanation of Drawings

A better understanding of the present invention can be obtained when the following detailed description of the preferred embodiment is considered inconjunction with the following drawings, in which:

Figure 1 is a block diagram of a system board of accmputer system according to the preferred embodiment; "

Figure 2 is a block diagram of a processor board according to the present invention for use with the computer system of Figure 1;

Figure 3 is a block diagram of a video system for use with the computer system of Figure 1;

Figure 4 is a block diagram of the memory controller of the processor board of Figure 2;

Figure 5 is a block diagram of the data buffer of Figure 2;

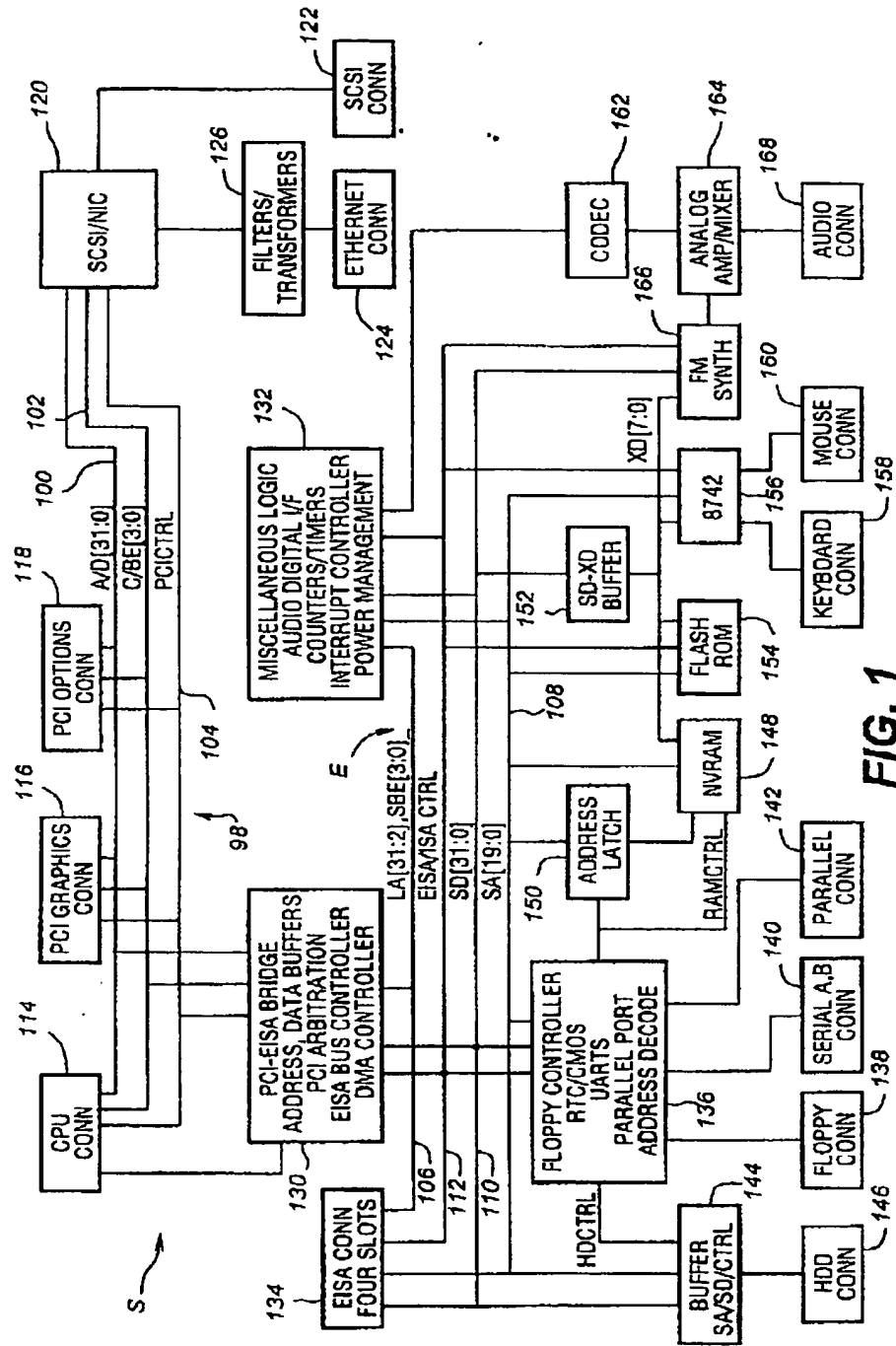
Figure 6 is a block diagram of the circuitry used to provide memory timing information and addresses for a particular memory cycle in the memory controller of Figure 4;

Figures 7 and 8 are block diagrams of portions of the memory controller of Figure 4;

Figure 9 is a block diagram of the CAM hit detection logic of the memory controller of Figure 4; and

Figures 10, 11, 12, 13, 14 and 15 are state machine diagrams for operations of circuitry of various portions of the memory controller of Figure 4.

【 Fig . 1 】



[Fig . 2]

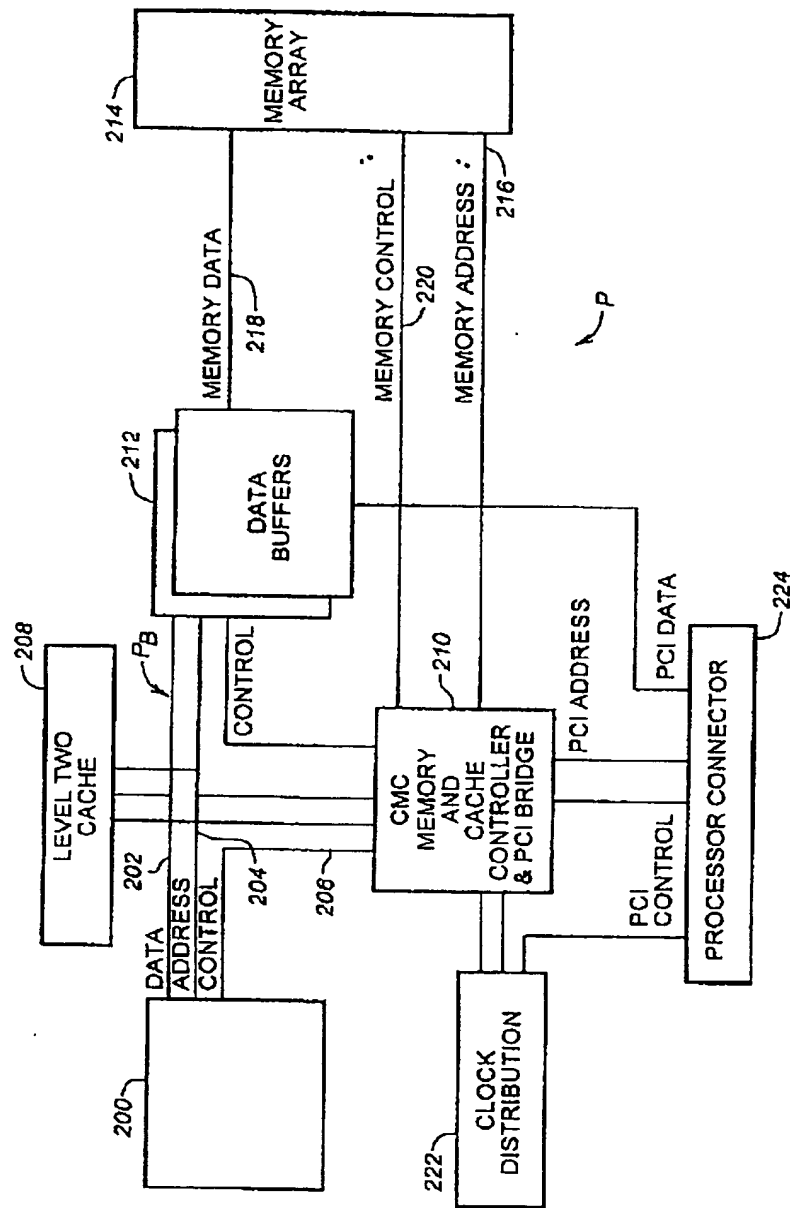


FIG. 2

【 F i g . 3 】

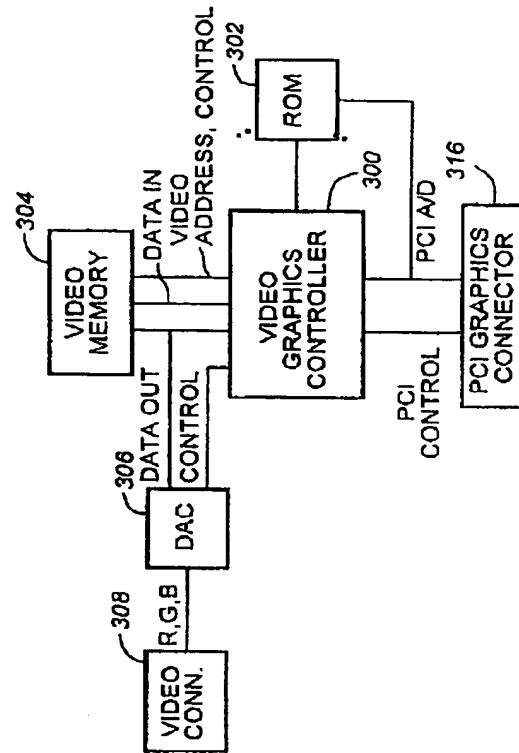


FIG. 3

【 Fig . 4 】

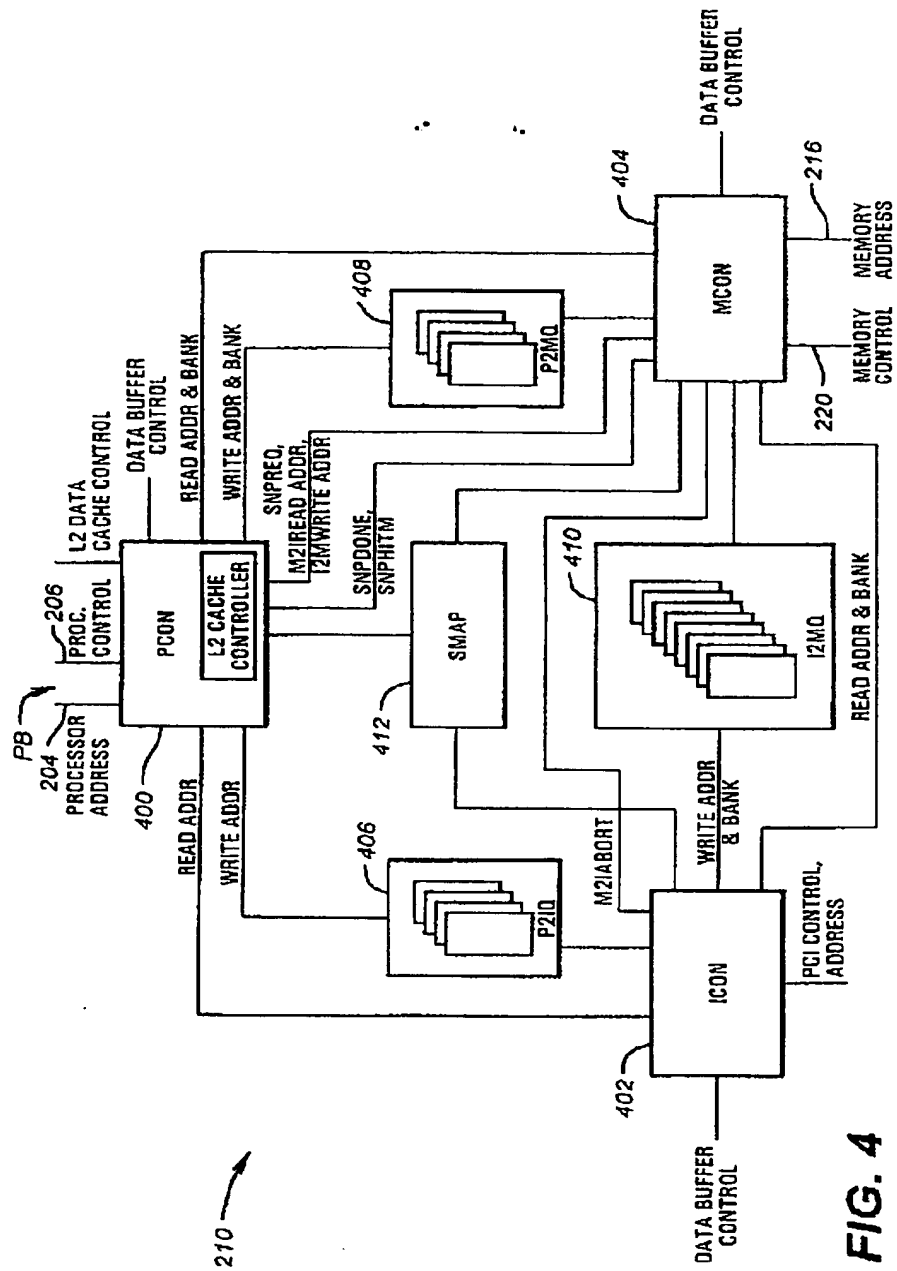
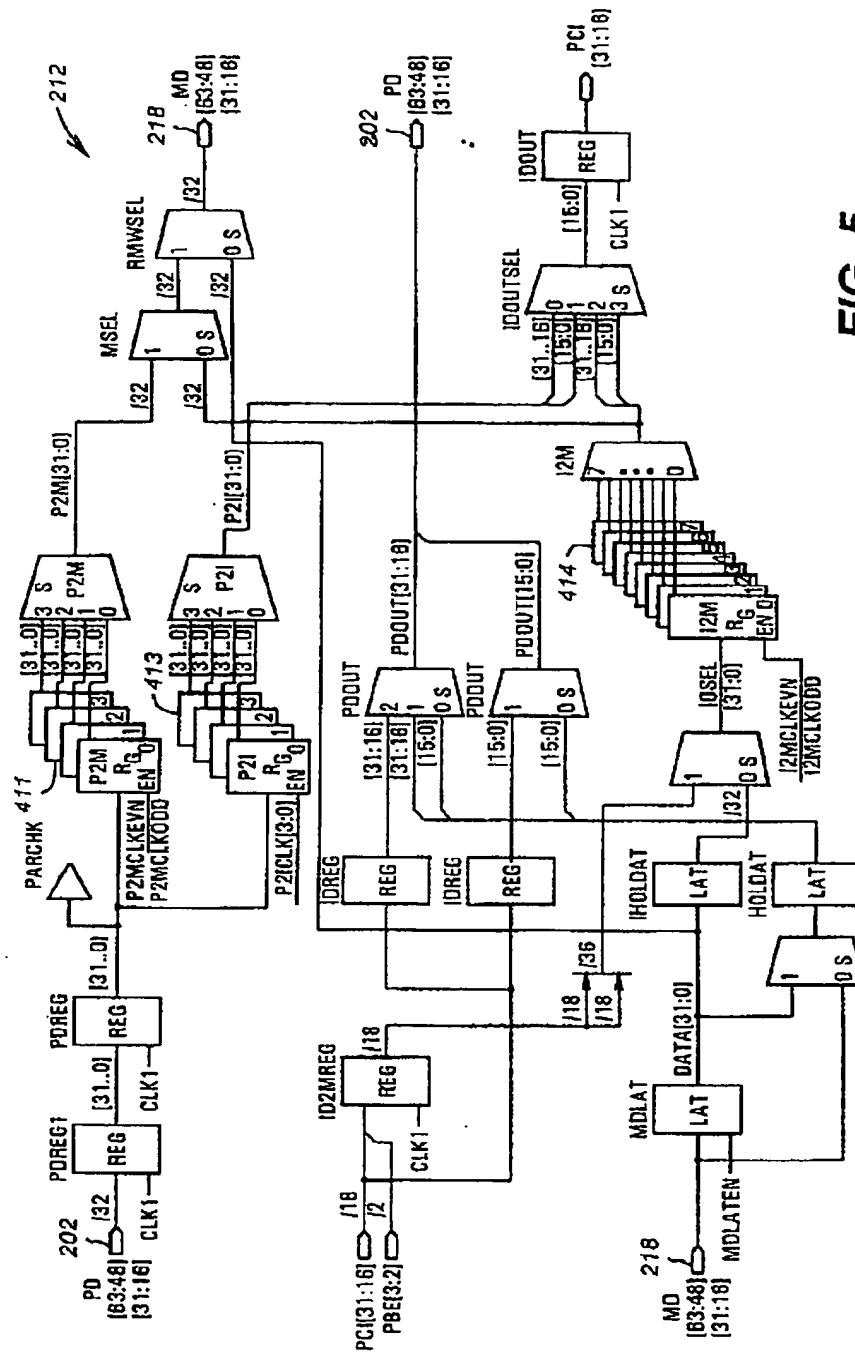


FIG. 4

FIG. 5



【 Fig . 6 】

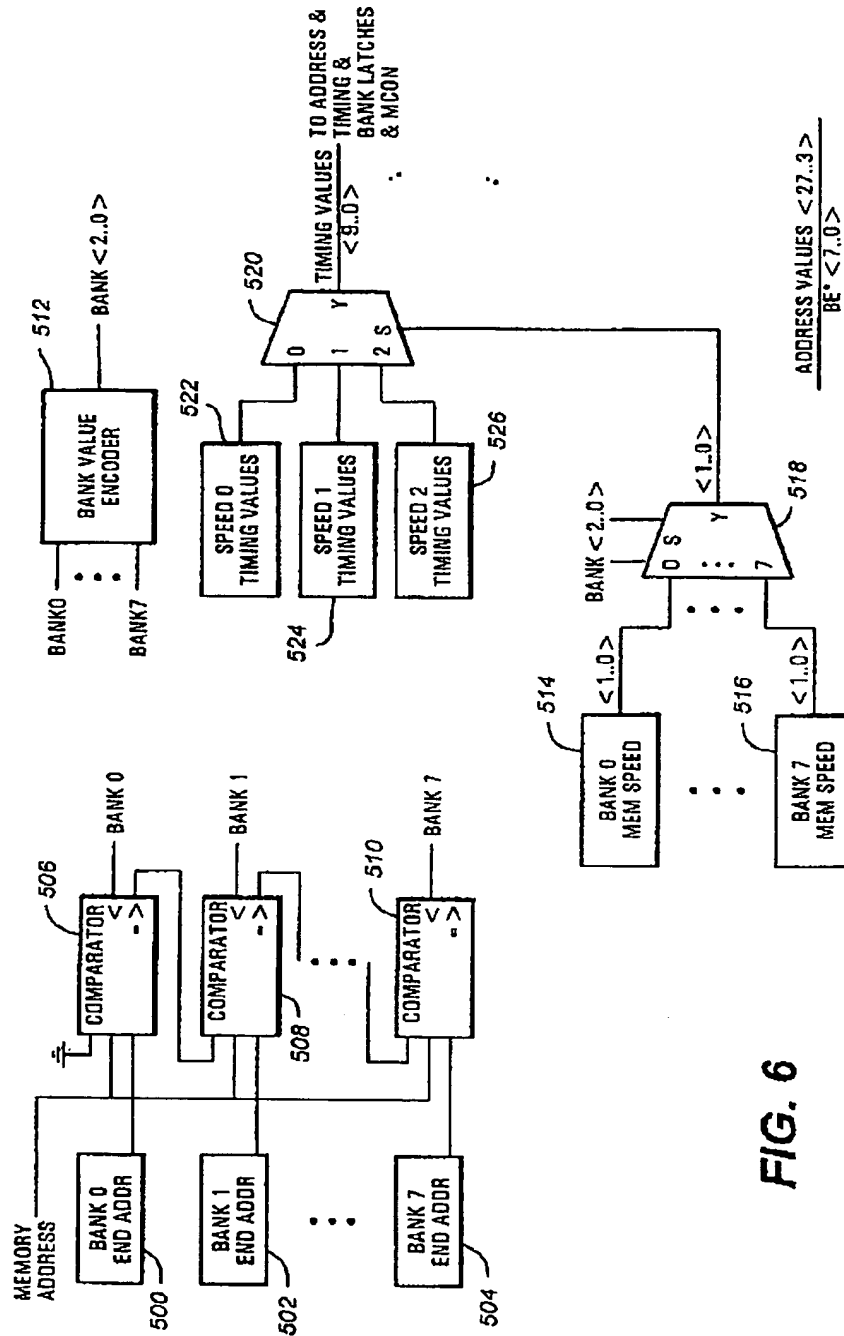
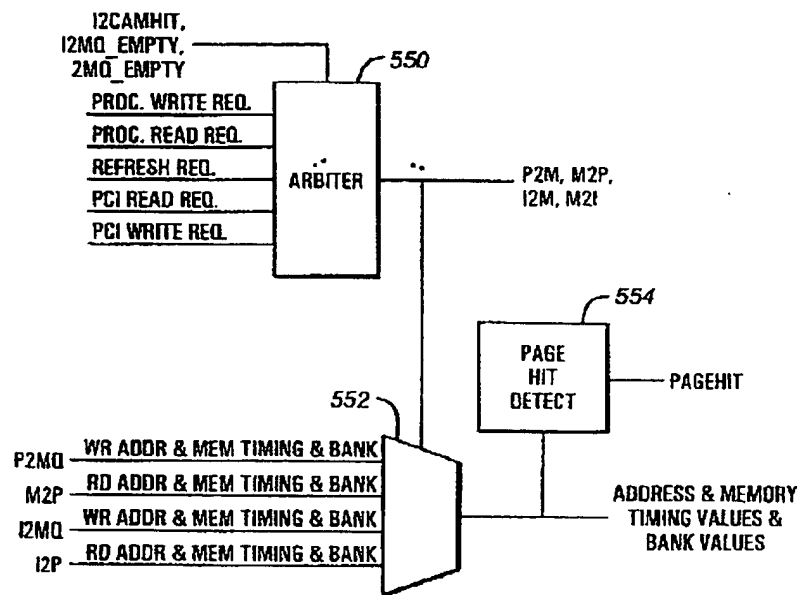


FIG. 6

【 Fig . 7 】

**FIG. 7**

【Fig. 9】

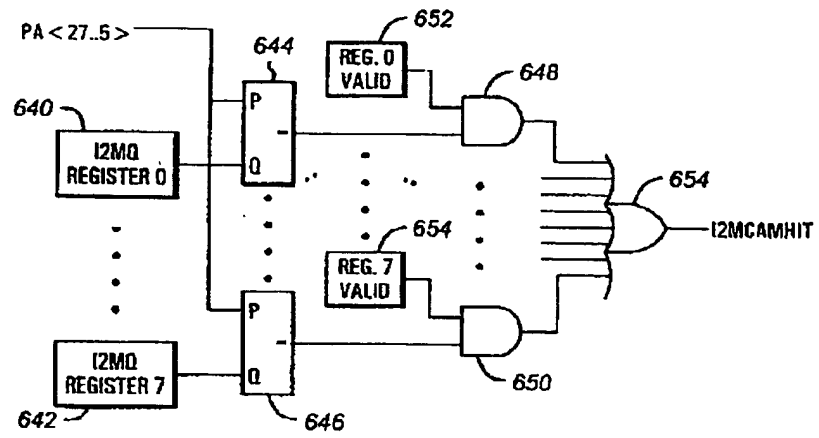
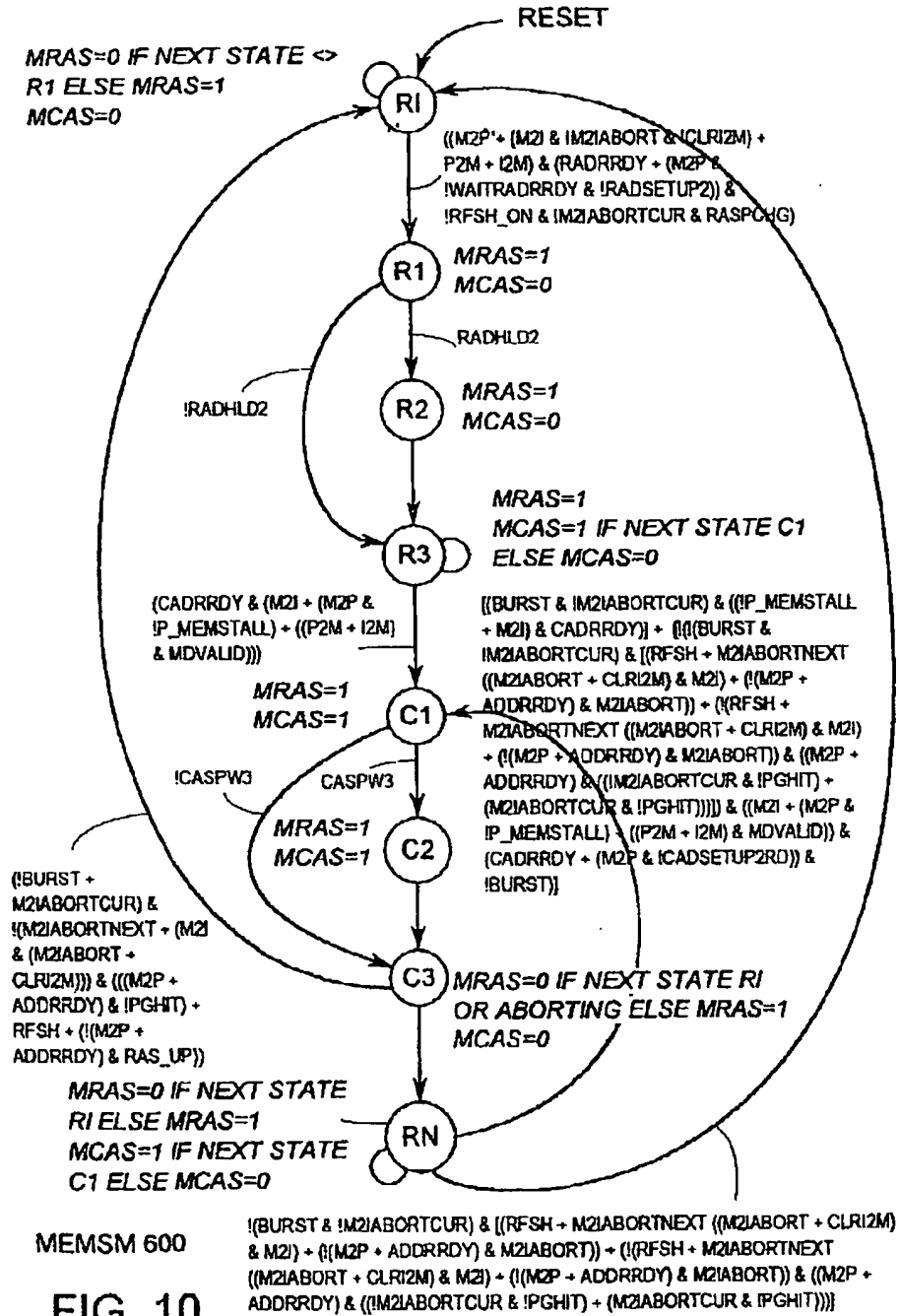
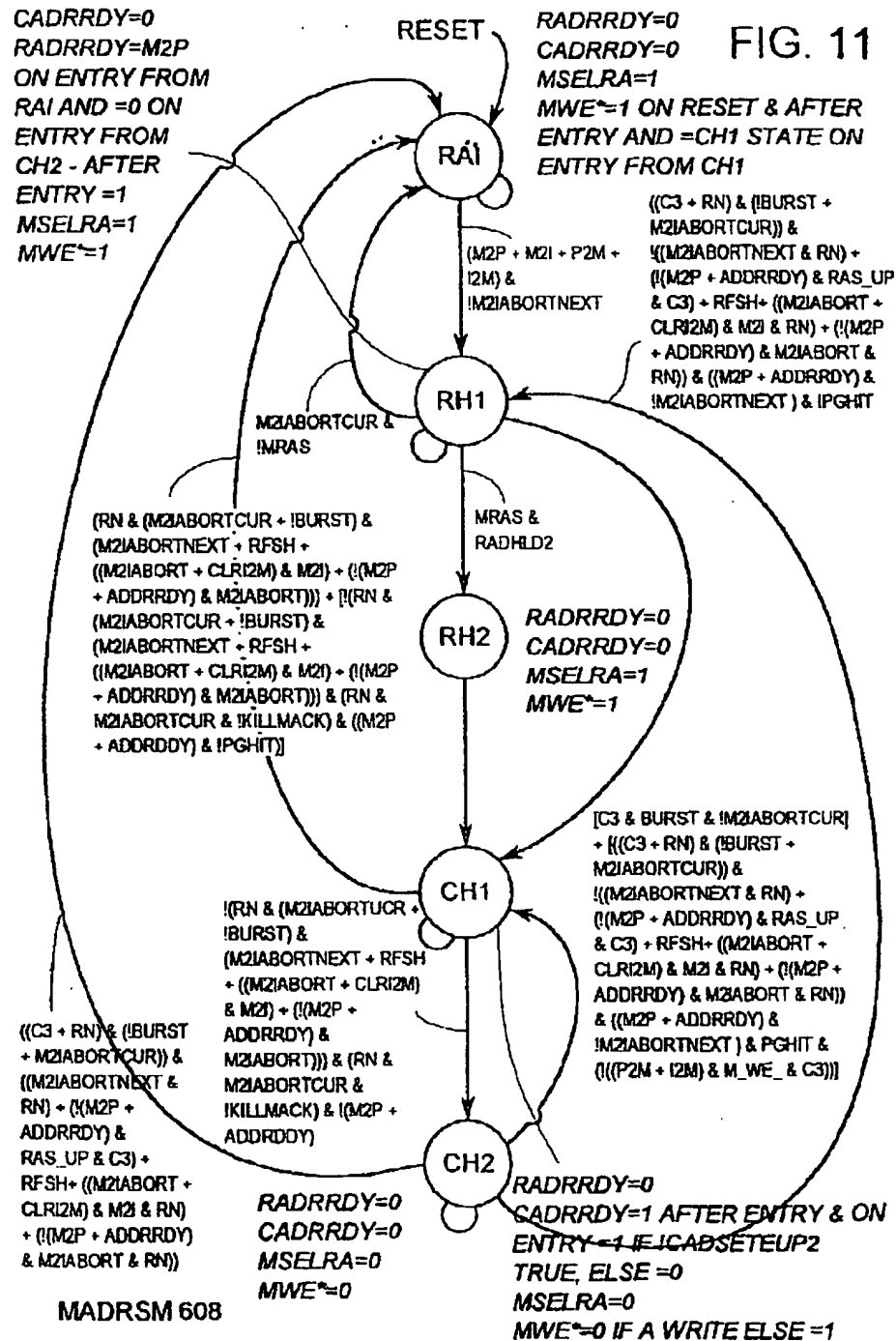


FIG. 9

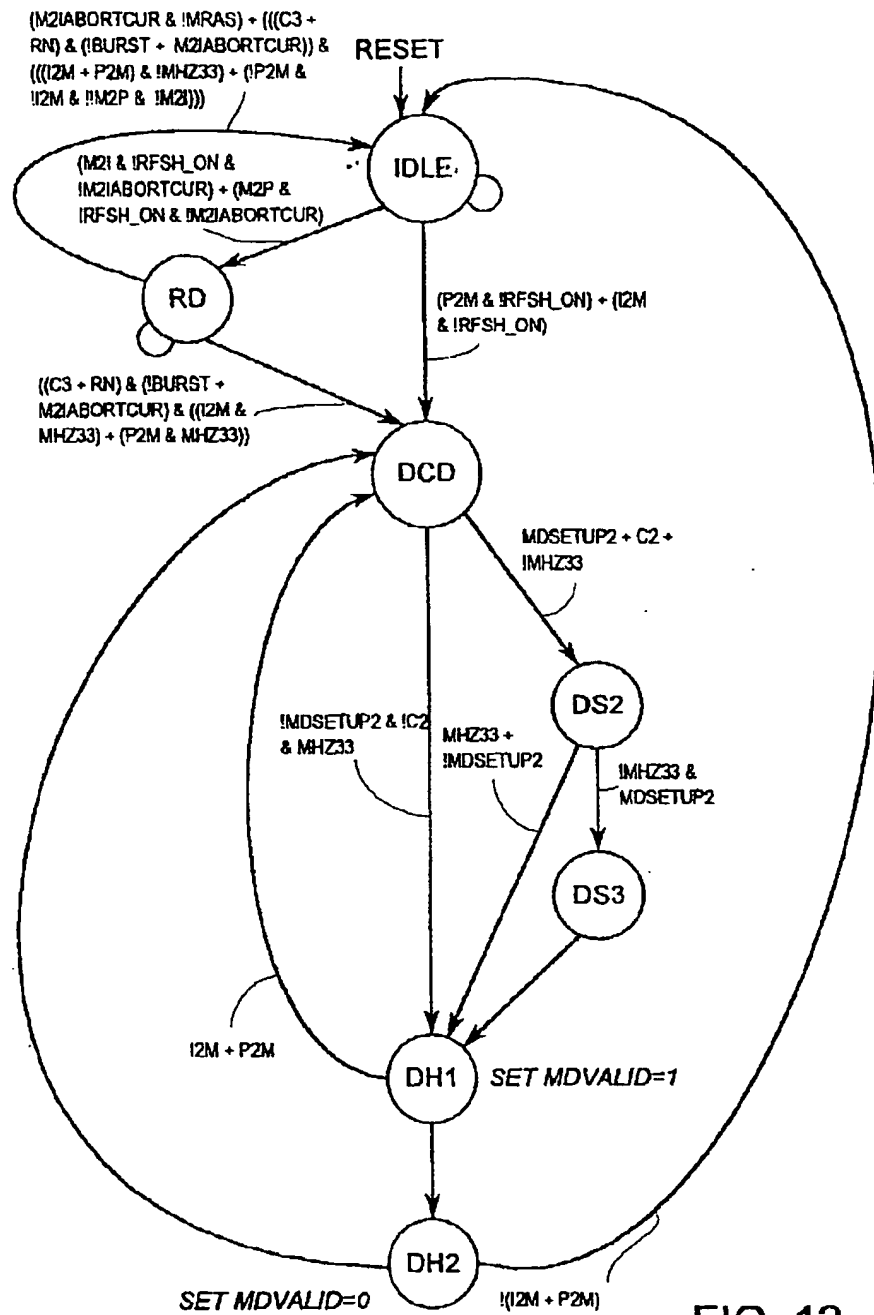
[Fig. 10]



【Fig. 11】



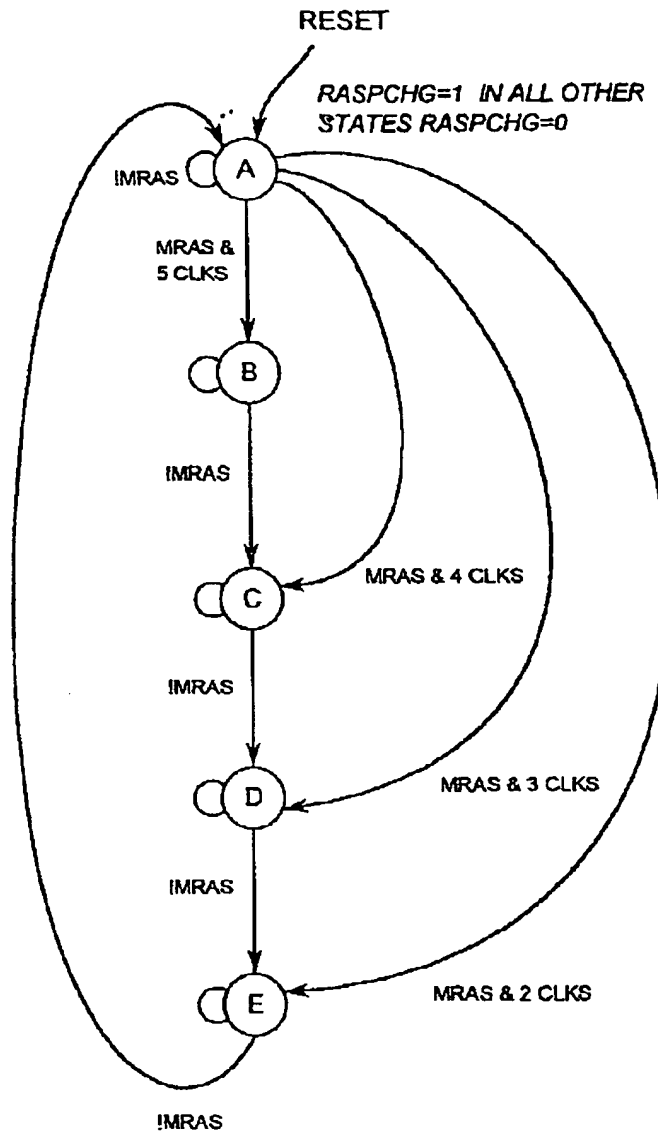
【 Fig . 1 2 】



MDCDSM 614

FIG. 12

【Fig. 13】



PRECHARGESM

FIG. 13

M2ISM

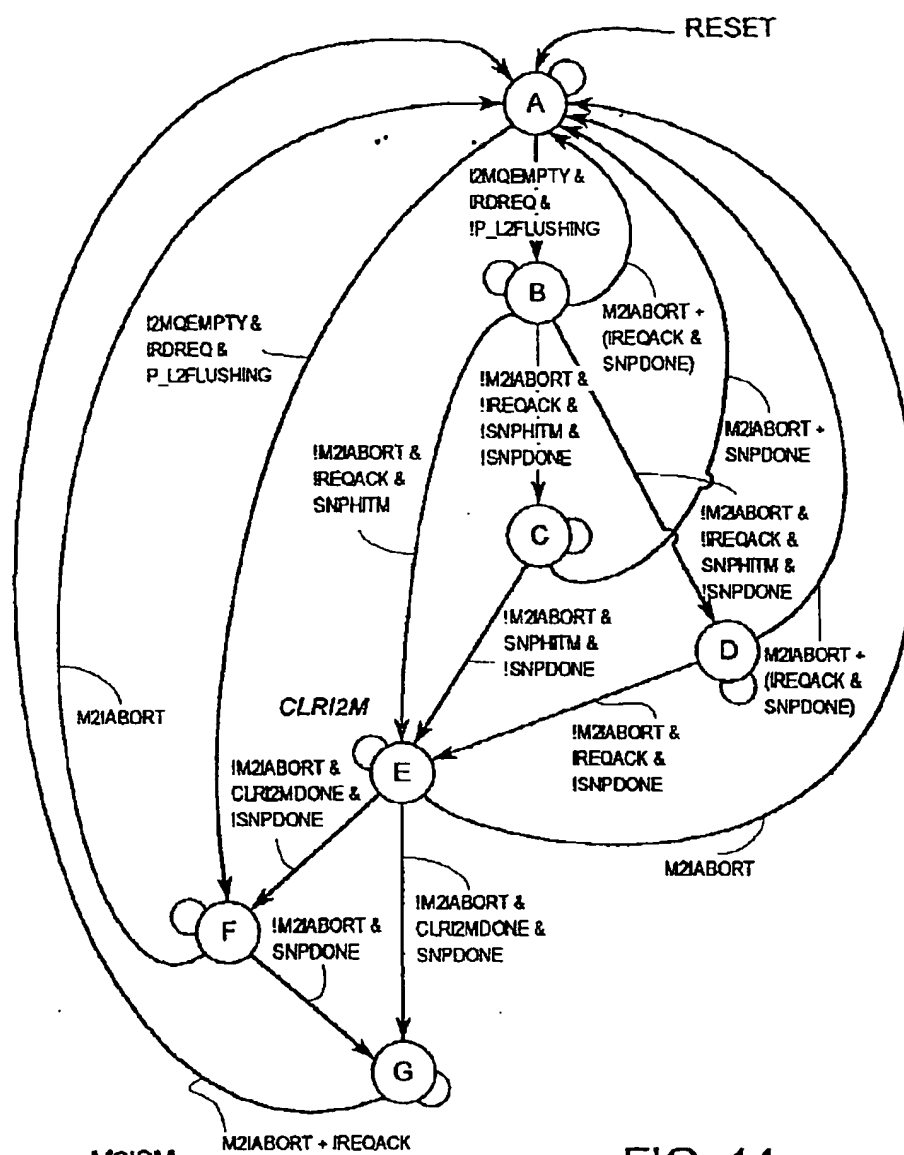
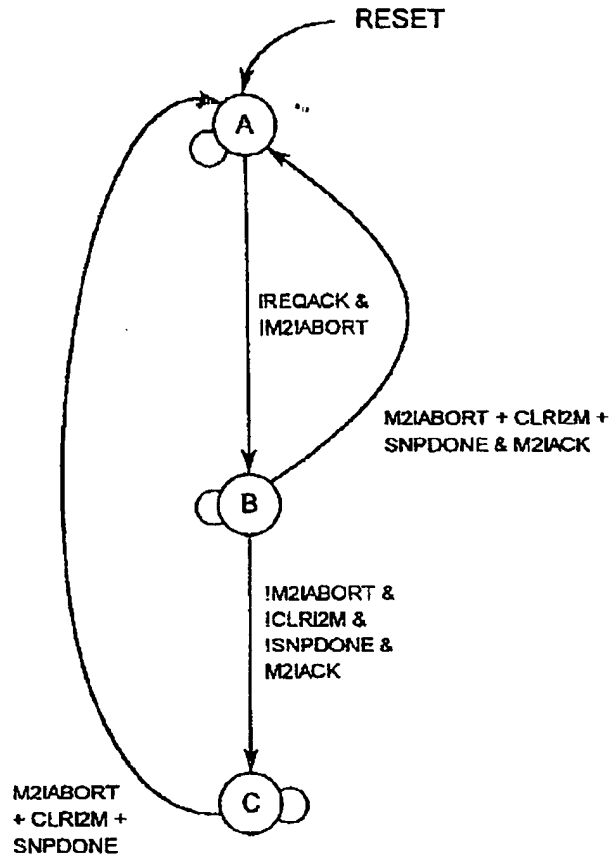


FIG. 14

【 F i g . 1 5 】



$$M2INA = !M2IABORT \& !CLR12M \& SNPDONE \& ((B \& M2IACK) + C)$$

M2INASM

FIG. 15

1 . Abstract

A memory controller which provides a series of queues between the processor and the PCI bus and the memory system. Memory coherency is maintained in two different ways. Before any read operations are accepted from the PCI bus, both of the posting queues must be empty. A content addressable memory (CAM) is utilized as the PCI to memory queue. When the processor performs a read request, the CAM is checked to determine if one of the pending write operations in the PCI to memory queue is to the same address as the read operation of the processor. If so, the read operation is not executed until the PCI memory queue is cleared of the write. To resolve the problem of aborting a Memory Read Multiple operation, an abort signal from the PCI bus interface is received and as soon thereafter as can be done the read ahead cycle is terminated, even though the read ahead cycle has not fully completed. The memory controller has improved prediction rules based on whether the cycle is coming from the processor or is coming from the PCI bus to allow more efficient precharging when PCI bus cycles are used. The memory controller is highly programmable for multiple speeds and types of processors and several speeds of memory devices. The memory controller includes a plurality of registers that specify number of clock periods for the particular portions of a conventional dynamic random access memory cycle which are used to control state machine operations.

2 . Representative Drawing

Fig . 1